



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **09198240 A**(43) Date of publication of application: **31.07.97**(51) Int. Cl. **G06F 9/06**(21) Application number: **08009616**(22) Date of filing: **23.01.96**(71) Applicant: **FUJITSU LTD**

(72) Inventor:  
**ENOKIDO SHIGENORI**  
**AKUTA HIROMI**  
**KIMURA TAKAYUKI**  
**KAMIOKA YASUSHI**  
**KATO KENJI**

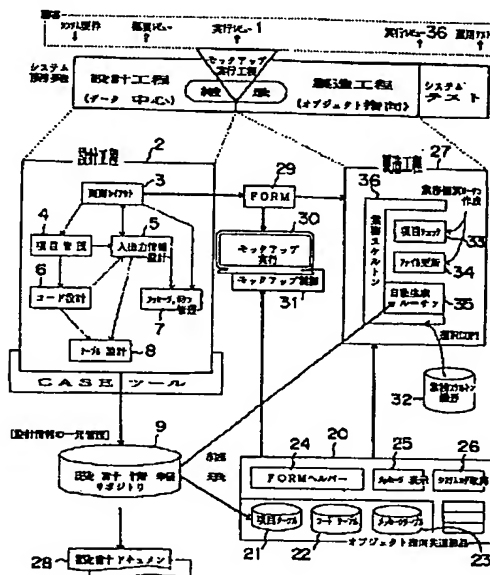
(54) **MOCK-UP METHOD AND CONTROLLER THEREFOR**

COPYRIGHT: (C)1997,JPO

## (57) Abstract:

**PROBLEM TO BE SOLVED:** To reduce the returns after a program is manufactured by performing the verification of a program and to efficiently perform the preparation of the program by taking over the data of a design process to a manufacturing process, by performing the same screen display as the screen display when the program is actually executed, in the program design process before the program manufacturing process of a job routine.

**SOLUTION:** A mock-up operation is performed by inputting the screen layout corresponding to a job routine in a FORM 29, preparing the item and code of the design information corresponding to the input of a screen by defining design information as a table, relating the location of the input of the screen and the record of the design information table before the program design process of a program manufacturing process, searching the design information table based on the input of the screen and displaying the output of the screen.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-198240

(43)公開日 平成9年(1997)7月31日

(51)IntCl.<sup>8</sup>  
G 0 6 F 9/06

識別記号 庁内整理番号  
5 3 0

F I  
G 0 6 F 9/06

技術表示箇所

5 3 0 T  
5 3 0 H

審査請求 未請求 請求項の数30 OL (全 42 頁)

(21)出願番号 特願平8-9616

(22)出願日 平成8年(1996)1月23日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72)発明者 榎戸 重則

東京都港区西新橋三丁目25番33号 株式会  
社富士通東京システムズ内

(72)発明者 阿久田 博美

東京都港区西新橋三丁目25番33号 株式会  
社富士通東京システムズ内

(74)代理人 弁理士 大曾 義之 (外1名)

最終頁に続く

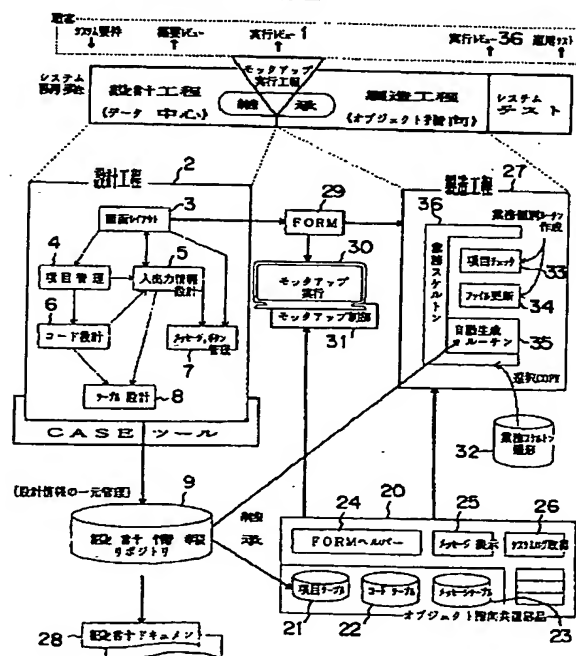
(54)【発明の名称】 モックアップ方法及びその制御装置

(57)【要約】

【課題】 業務ルーチンのプログラム製造工程以前のプログラム設計工程において、実際にプログラムが実行されたと同様な画面表示を行うことにより、プログラムの検証を行うことによりプログラム製造後の手戻りを減少させるとともに、設計工程のデータを製造工程に継承することにより、プログラム作成の効率化を図る。

【解決手段】 業務ルーチンに対応した画面レイアウトをFORMに入力させるとともに、画面入力に対応した設計情報の項目とコードとを設計情報をテーブルとして作成し、プログラム製造工程のプログラム設計工程以前において、画面入力の位置と設計情報テーブルのレコードとを関連付け、画面入力に基づいて設計情報テーブルをサーチして画面出力を表示することによりモックアップ動作を行う。

本発明の原理説明図



## 【特許請求の範囲】

【請求項1】 業務ルーチンに対応する画面レイアウトをFORMに入力するステップと、

FORMに入力されるべき項目情報の設計情報を業務ルーチンの製造工程以前の段階において作成するステップと、

前記業務ルーチンの製造工程以前の段階で、前記項目情報の画面への入力に対応して前記項目情報に関する情報を前記設計情報に基づいて画面に表示し、実際に業務ルーチンが実行された場合の画面表示を検証するモックアップ実行ステップとからなることを特徴とするモックアップ方法。

【請求項2】 前記モックアップ実行ステップはFORMへの入力の画面上のフィールドに基づいて前記設計情報をサーチして設計情報に対応する出力を画面に表示するFORMヘルパーステップからなることを特徴とする請求項1記載のモックアップ方法。

【請求項3】 前記FORMヘルパーステップは、設計情報に含まれる項目情報を前記画面上のフィールドに対応したタグ領域にもセットすることを特徴とする請求項2記載のモックアップ方法。

【請求項4】 前記FORMヘルパーステップは、画面上のフィールドに対応したタグ領域にセットされた項目情報と、設計情報に含まれる項目情報とに基づいて画面上の入力と設計情報とを関連付けて前記入力に対応する出力を画面表示することを特徴とする請求項3記載のモックアップ方法。

【請求項5】 前記FORMヘルパーステップは、前記FORMに入力される所定項目情報の画面上の位置と前記所定項目情報の前記設計情報が格納されたファイルのアドレスとを関連付けることにより、前記FORMに入力された所定項目情報の正当性を検証することを特徴とする請求項4記載のモックアップ方法。

【請求項6】 前記FORMヘルパーステップは、前記FORMに入力された所定項目情報に対応する項目IDと、

前記設計情報の項目IDとを用いて両者の関連付けを行うことを特徴とする請求項5記載のモックアップ方法。

【請求項7】 前記設計情報を作成するステップは、項目ディクショナリを用いて、前記FORMに入力された所定項目に対応する項目IDを有する入出力情報定義体を作成するステップと、

入出力情報定義体を用いて、前記FORMに入力された所定項目に対して対応する項目IDをペーストするステップからなることを特徴とする請求項1記載のモックアップ方法。

【請求項8】 前記FORMヘルパーステップは、前記FORMに入力された所定項目情報に関する設計情報を有する設計情報テーブルを作成するステップと、

FORMに入力された項目情報の正当性を前記設計情報テーブルを用いて判断するチェックステップからなることを特徴とする請求項2記載のモックアップ方法。

【請求項9】 前記設計情報テーブルは項目情報が記憶される項目テーブルと項目情報の中の具体的情報を含むコードテーブルの少なくとも1つからなることを特徴とする請求項8記載のモックアップ方法。

【請求項10】 前記チェックステップは、前記項目テーブルからデータが入力されたフィールドの設計桁数を取り出すステップと、入力されたデータのデータ長を調べるステップと、前記設計桁数とデータ長とを比較するチェック工程とからなる桁数チェックルーチンを行うことを特徴とする請求項9記載のモックアップ方法。

【請求項11】 前記チェックステップは、前記項目テーブルからデータが入力されたフィールドの設計属性を取り出すステップと、入力されたフィールドについてその属性に基づいて数字チェック、英数字チェック或いは日付チェックのうちの少なくとも1つを行う工程とからなる属性チェックステップを行うことを特徴とする請求項9記載のモックアップ方法。

【請求項12】 前記チェックステップは、前記項目テーブルからデータが入力されたフィールドの必須桁数を取り出すステップと、入力されたデータのデータ長を調べるステップと、必須桁数がデータ長より大きくない場合に必須項目に入力がされたものとする必須入力チェック工程を有することを特徴とする請求項9記載のモックアップ方法。

【請求項13】 前記コードテーブルを用いて画面の特定のフィールドに対応するコードと名称を参照するか又は特定のフィールドに入力されたコードのチェックを行うことを特徴とする請求項9記載のモックアップ方法。

【請求項14】 前記コード参照ステップは画面の特定のフィールドから項目内アドレスを基にTAGと画面テーブルを用いて項目テーブルを参照し、その項目テーブル内のコードIDをひき、コードIDに基づいてコードテーブルを参照し、そのコードテーブルから所望のコードと名称を選択することを特徴とする請求項13記載のモックアップ方法。

【請求項15】 前記コードチェックステップはコードが入力された画面の特定のフィールドの項目内アドレスを基にTAGと画面テーブルを用いて項目テーブルを参照し、その項目テーブル内のコードIDをひき、コードIDに基づいてコードテーブルを参照し、そのコードテーブルから入力されたコードがあるか否かを判断し、あればその名称を表示することを特徴とする請求項13記載のモックアップ方法。

【請求項16】 前記モックアップ実行ステップは、項目テーブルから今までカーソルがあった項目の編集1

Dを取り出すステップと、  
その編集IDが存在する場合に画面上のテキストエリアをセーブするステップと、  
テキストエリアにセーブした内容から数値編集、通貨編集、或いは日付編集のうちの少なくともどれか1つの編集を行うステップとからなる項目編集工程を有することを特徴とする請求項2記載のモックアップ方法。

【請求項17】 メッセージ、ボタン情報を作成するステップを更に有し、  
前記モックアップ実行ステップにおいて、前記メッセージ情報、及びボタン情報を画面に表示することを特徴とする請求項2記載のモックアップ方法。

【請求項18】 前記FORMと項目情報とメッセージボタン情報との少なくとも1つを製造工程に継承することを特徴とする請求項17記載のモックアップ方法。

【請求項19】 画面に対応するFORMと、  
業務ルーチンに係る業務上のコードを少なくとも含むコードテーブルと、  
少なくとも項目IDとコードIDを含む項目テーブルと、

前記FORM上のフィールドの位置と項目テーブルのアドレスとを関連付けるポインタ手段と、

前記ポインタ手段を用いて前記コードテーブルからコードと名称を参照するかまたは前記FORMのフィールドに入力したコードをもとにコードが正しいかどうかをチェックし、前記コードおよび前記名称を前記FORMの特定のフィールドに表示する名称参照及びコードチェック手段とからなり、

前記名称参照及びコードチェック手段を用いて、前記業務ルーチンに係る前記項目テーブルとそれに対応する前記コードテーブルを前記業務ルーチンの実際のプログラム作成とは独立して作成することを特徴とするモックアップ制御装置。

【請求項20】 前記モックアップ制御装置を構成するFORM、コードテーブル及び項目テーブルを設計工程から自動継承したことを特徴とする請求項19記載のプログラム製造装置。

【請求項21】 画面に対応するFORMと、  
項目IDとコードIDを含む項目テーブルと、  
前記FORM上のフィールドのアドレスと項目テーブルのアドレスとを関連付けるポインタ手段と、  
前記FORMに対する入力工程、前記項目テーブルを参照して行う工程に伴う画面上の遷移を画面をヘッド、ボディ、テイルとグループ化して順次行うグループ制御手段と、

前記グループ制御手段を用いて、前記業務ルーチンの少なくとも一部と同様の画面遷移を実際のプログラム作成とは独立して実行するモックアップ制御装置。

【請求項22】 業務ルーチンに用いられるデータと関係する画面を入力する手段と、

前記業務ルーチンに用いられるデータに関する設計情報を格納する手段と、

前記画面のデータの位置と前記設計情報とを関連させる手段とからなるプログラム作成装置。

【請求項23】 前記関連手段は、業務ルーチンの製造以前の設計段階において、

前記画面に業務ルーチンに用いられるデータを入力したとき前記設計情報に基づいてかかるデータ入力に対応した出力結果を画面に表示することを特徴とする請求項22記載のプログラム作成装置。

【請求項24】 業務ルーチンの製造以前の設計段階において、

前記画面に入力された項目情報に対応して作成された設計情報のうち、用語ディクショナリ、項目ディクショナリ、入出力情報定義体、項目テーブル、メッセージテーブル、ボタンテーブルのうちの少なくとも1つを業務ルーチンの製造段階でもそのまま自動継承することを特徴とする請求項23記載のプログラム作成装置。

【請求項25】 業務ルーチンに用いられるフォームを画面に入力するステップと、

プログラム製造段階の前の設計段階において、該フォームへの入力に対応して設計データファイルを作成するステップと、

フォームに入力した項目情報の正当性を前記設計情報に基づいて制御し、その結果を画面に表示するステップと、

前記設計情報をプログラム製造工程に自動継承するステップとよりなるプログラム作成方法。

【請求項26】 業務ルーチンに対応してFORMを作成するステップと、

少なくとも業務に使用される項目の項目IDを含む項目テーブルを作成するステップと、

前記FORM上の所定項目に対してTAG領域を設けるステップと、

前記TAG領域に前記FORMの所定項目に対応する項目IDをセットするステップとからなるFORMヘルパー設計方法。

【請求項27】 請求項26記載のFORMヘルパー設計方法で設計されたFORMヘルパーを用いて、

前記項目テーブルのアドレスごとにアドレスが設けられ画面テーブルを作成するステップと、

前記TAG領域にセットされた項目IDに基づいて前記項目テーブルをサーチするステップと、

項目IDがマッチングした場合、前記TAG領域に項目テーブルの前記項目のアドレスをセットすると共に画面テーブルに、前記FORM上の所定項目のフォーム内項目アドレスをセットするステップからなることを特徴とするFORMヘルパー制御方法。

【請求項28】 請求項27記載のFORMヘルパー制御方法を用いて、

FORMに業務ルーチンに対応する所定項目情報を入力するステップと、

前記指定項目の設計情報に基づいて入力された所定項目情報に対応する出力を画面表示するステップと、

前記画面入力ステップ及び画面出力ステップを業務ルーチンの製造工程前に行うことにより、実際に業務ルーチンが製造された場合の画面表示を検証するステップとからなることを特徴とするモックアップ方法。

【請求項 29】 前記項目テーブルを作成するステップにおいて、項目テーブルのアドレスに対応してソースプログラム作成時に視認性のあるコードデータを格納する項目インデックス格納ファイルを作成することを特徴とする請求項 26 記載の FORM ヘルパー設計方法。

【請求項 30】 業務ルーチンに対応して画面上で用いられる項目の一部を構成する用語とその ID を格納する用語ディクショナリを作成するステップと、

前記用語を組み合わせる項目を作成し、用語 ID を組み合わせる項目 ID を作成し、この項目を項目 ID を対応させて項目ディクショナリを作成するステップとからなることを特徴とする項目ディクショナリ作成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は画面制御モックアップシステムに係り、特にソフトウェアシステムを提供するにあたって、ソフトウェアプログラムを最後まで製造する前に設計から製造への途中の段階でソフトウェアの擬似動作を画面上に表すためのモックアップ（模型）を提供することにより、ソフトウェアの設計・製造工程を効率化した画面制御モックアップシステムに関する。

【0002】

【従来の技術】従来ソフトウェアの設計を行うにあたってはユーザよりソフトウェアが提供されるべきシステムの説明をシステムエンジニアが理解し、そのユーザのシステムに沿った形にソフトウェアの基本設計作業を行い、その後実際にプログラムのコーディングを行った後、ユーザに再びそのソフトウェアを提供しユーザが希望するシステムの通り動作するか否かを検討する。したがって、実際にプログラムがコーディングまでされた後において、その出来上がったソフトウェアシステムが、ユーザの希望するものと異なっていた場合はコーディングまで完了した時点で再びフィードバック（手戻り）となって設計の段階からの出直しを行わなければならなかった。

【0003】更にユーザにとってみれば、メーカーに依頼したソフトウェアが実際に自分の希望通り動作するか否かについての検証を行うまでかなりの時間を要しているので、ソフトウェアが実際にコーディングされるまでの間待たされ続けるということがあった。

【0004】

【発明が解決しようとする課題】近年、上記の点に鑑み

ていわゆるスパイラル方式あるいはプロトタイプ方式と呼ばれる手法が提案されている。この手法によれば、ソフトウェアを設計段階から製造段階までウォーターフォール形式に作成して仕舞わずに、その工程の途中において SE あるいはソフトウェアメーカーはユーザに対して頻繁にこの経過を報告し必要においては途中までコーディングされたものをユーザに検証してもらうことにより実際にプログラムの製造完了した後にフィードバックをかけることから生じる上述した問題を解決しようとしていた。しかしながら、この手法もソフトウェアメーカーがユーザと再三にわたり打ち合わせをする度に、コーディングの手戻りが生じ、工程が完了した後でコーディングを変更する場合と比べて全体のソフトウェア制作工程に使用する時間はそれほど効率が上がらないことが現実であった。

【0005】さらに、従来はユーザからシステム要件を入手した SE は、まず紙かあるいはワードプロセッサで業務ルーチンの実行に用いられる画面レイアウトを作成し、しかるのち業務ルーチンの設計を紙やワードプロセッサで作成した画面レイアウトに従って行っていたので、設計を途中で変更した場合、紙かワードプロセッサで作成された画面レイアウトは変更されない状態に残ってしまうという問題があった。

【0006】また、設計工程が終了後、製造工程を何も無いところから開始しなければならないので製造工程に時間がかかるという問題もあった。

【0007】

【課題を解決するための手段】本発明は上記従来の問題点に鑑みて、ソフトウェアの設計工程から製造工程まで作業を効率的にし、かつユーザの希望を具体的に反映したソフトウェアをより迅速に完成するソフトウェア開発手法を提供することを目的とする。

【0008】また、本発明では設計工程で画面レイアウトに表示すべき項目情報を作成し、その設計工程で作成されたデータを製造工程でも継承し、製造工程をより短縮化することを目的とする。

【0009】さらに、本発明では SE がシステム要件を入手したとき設計工程を直接 FORM に入力することから設計工程を開始することにより画面レイアウト等のデータを一元化することを目的とする。

【0010】本発明の一態様は、業務ルーチンに対応する画面レイアウトを FORM に入力するステップと、FORM に入力されるべき項目情報の設計情報を業務ルーチンの製造工程以前の段階において作成するステップと、前記業務ルーチンの製造工程以前の段階で、前記項目情報の画面への入力に対応して前記項目情報に関する情報を前記設計情報に基づいて画面に表示し、実際に業務ルーチンが実行された場合の画面表示を実現するモックアップ実行ステップとからなるモックアップ方法を提供することである。

【0011】本発明の他の態様は、画面に対応するFORMと、業務ルーチンに係る業務上のコードを少なくとも含むコードテーブルと、少なくとも項目IDとコードIDを含む項目テーブルと、前記FORM上のフィールドの位置と項目テーブルのアドレスとを関連付けるポインタ手段と、前記ポインタ手段を用いて前記コードテーブルからコードと名称を参照するかまたは前記FORMのフィールドに入力したコードをもとにコードが正しいかどうかをチェックし、前記コードおよび前記名称を前記FORMの特定のフィールドに表示する名称参照及びコードチェック手段とからなり、前記名称参照及びコードチェック手段を用いて、前記業務ルーチンに係る前記項目テーブルとそれに対応する前記コードテーブルを前記業務ルーチンの実際のプログラム作成とは独立して作成するモックアップ制御装置を提供することである。

【0012】本発明の他の態様は、画面に対応するFORMと、項目IDとコードIDを含む項目テーブルと、前記FORM上のフィールドのアドレスと項目テーブルのアドレスとを関連付けるポインタ手段と、前記FORMに対する入力工程、前記項目テーブルを参照して行う工程に伴う画面上の遷移を画面をヘッド、ボディ、テイルとグループ化して順次行うグループ制御手段と、前記グループ制御手段を用いて、前記業務ルーチンの少なくとも一部と同様の画面遷移を実際のプログラム作成とは独立して実行するモックアップ制御装置を提供することである。

【0013】本発明の他の態様は、業務ルーチンに用いられるデータと関係する画面を入力する手段と、前記業務ルーチンに用いられるデータに関する設計情報を格納する手段と、前記画面のデータの位置と前記設計情報とを関連させる手段とからなるプログラム作成装置を提供することである。

【0014】本発明の他の態様は、業務ルーチンに用いられるフォームを画面に入力するステップと、プログラム製造段階の前の設計段階において、該フォームへの入力に対応して設計データファイルを作成するステップと、フォームに入力した項目情報の正当性を前記設計情報に基づいて制御し、その結果を画面に表示するステップと、前記設計情報をプログラム製造工程に自動継承するステップとよりなるプログラム作成方法を提供することである。

【0015】本発明の他の態様は、業務ルーチンに対応してFORMを作成するステップと、少なくとも業務に使用される項目の項目IDを含む項目テーブルを作成するステップと、前記FORM上の所定項目に対してTAG領域を設けるステップと、前記TAG領域に前記FORMの所定項目に対応する項目IDをセットするステップとからなるFORMヘルパー設計方法を提供することである。

【0016】本発明の他の態様は、FORMヘルパー設計方法で設計されたFORMヘルパーを用いて、前記項目テーブルのアドレスごとにアドレスが設けられ画面テーブルを作成するステップと、前記TAG領域にセットされた項目IDに基づいて前記項目テーブルをサーチするステップと、項目IDがマッチングした場合、前記TAG領域に項目テーブルの前記項目のアドレスをセットすると共に画面テーブルに、前記FORM上の所定項目のフォーム内項目アドレスをセットするステップからなることを特徴とするFORMヘルパー制御方法を提供することである。

【0017】本発明の他の態様は、FORMヘルパー制御方法を用いて、FORMに業務ルーチンに対応する所定項目情報を入力するステップと、前記指定項目の設計情報に基づいて入力された所定項目情報に対応する出力を画面表示するステップと、前記画面入力ステップ及び画面出力ステップを業務ルーチンの製造工程前に行うことにより、実際に業務ルーチンが製造された場合の画面表示を検証するステップとからなることを特徴とするモックアップ方法を提供することである。

【0018】本発明の他の態様は、業務ルーチンに対応して画面上で用いられる項目の一部を構成する用語とそのIDを格納する用語ディクショナリを作成するステップと、前記用語を組み合わせて項目を作成し、用語IDを組み合わせて項目IDを作成し、この項目を項目IDに対応させて項目ディクショナリを作成するステップとからなる項目ディクショナリ作成方法を提供することである。

【0019】本発明によれば、設計工程において、まず画面に対応するFORMに業務ルーチンに用いられる項目情報を打ち込み、その項目情報を項目ディレクトリに登録し、画面に項目名等の必要な項目データを選択表示することに伴って、画面表示に対応した入出力情報定義体を生成するものである。入出力情報定義体は画面上の表示と一対一に対応しているので、画面上の表示の制御を行うにあたって、コンピュータ内部では入出力情報定義体を参照しながら制御を行えばよいことになる。そして、データエントリプログラムを検証するに際しては、画面に表示されたデータ、例えば処理区分等の項目が設計された属性に合っているか否かをチェックルーチンによって入出力定義体を参照することにより検出することができる。例えば、処理区分の項目が8桁のデータよりなるものと属性が定められた場合にあって、6桁のデータを画面上に入力した場合には、その入力はエラーであることを画面上に表示する。したがって、ユーザは自分が入力した処理区分のデータが正しく入力されたものであるか否かを実際のプログラムが製造される以前に知ることができる。

【0020】したがってユーザは、例えば処理区分等のデータのエントリを、実際にプログラムがコーディング

され、製造される以前に本発明において提供されたモックアップシステムを用いることにより確認することができる。このため、ソフトウェアメカにとってはデータ入力の状態を実際にプログラムをコーディングすることなく、擬似的に画面上で表示あるいは動作することができるので、いちいちデータエントリプログラムを製造してから検証し、手戻りをしてコーディングを訂正していた場合に比べてプログラム作成工程を著しく簡略化できる。さらにソフトウェアのユーザにとっては実際にプログラムがコーディングされるまで待たされることなく、極めて早い段階において、必要においてはソフトウェアユーザがソフトウェアメカに対して自分の希望するシステムを伝達すると直ちに前記モックアップが動作し本システムが動作することによって擬似出力を画面上に表示することができるので、そして自分のデータエントリの手順が正しいか否か或いは自分の自らの説明したシステムがプログラムを経た後で画面上にいかなる形で展開されるかを実際にプログラムが製造される以前の段階で、すなわち極めて早い段階において確認することができるものである。

【0021】さらに上述のごとくモックアップ動作に用いられたプログラムエントリの工程は、そのままプログラムの製造工程に自動継承することができるので、モックアップ動作によって生成された部分については、実際のプログラム製造段階では改めてプログラムする必要がないため、プログラムの実際の製造工程を高速化することが可能である。

【0022】更に、データエントリを行うに際しては、画面制御等の共通部品あるいはメッセージ等の業務共通ルーチンは殆どのプログラムにおいても共通に使用されるものであるので、かかる共通部品あるいは業務共通ルーチンはこのモックアップ工程において再利用可能であり、さらにこの改良された共通部品あるいは業務共通ルーチンは製造工程においてもそのまま自動継承されて再利用可能になるのである。必要に応じては業務ルーチンにおいても、業務スケルトン部分はモックアップ動作と関連して使用することができ再利用可能となる。

【0023】さらに、本発明では、設計工程におけるデータを一元化したので、設計変更に伴うデータ修正を簡単にできる。以下、図1を参照して本発明の原理を具体的に説明する。

【0024】顧客からシステム要件を入手したSE（システムエンジニア）は、システム開発を開始し、データ中心の設計工程を行う。この設計工程の途中で顧客がプログラムの概要レビューを行うが、この段階ではSEの設計した画面を顧客は見るだけであり、その画面に動きはない。

【0025】次に、本発明のモックアップ実行工程に移るとまだプログラムが製造されていない段階にも係わらず、プログラムが実行されているように画面表示を制御

し、モックアップ実行レビュー1を行うことができる。このモックアップ実行レビューにより実際にプログラムを製造して動かした場合に問題となることを顧客から指摘されれば、SEはプログラムの設計を修正することが可能である。すなわち、設計工程作業2でCASEツールを用いて作成された画面レイアウト3、項目管理情報4としての入出力情報設計5及びコード設計6、メッセージ、ボタン管理情報7、及びテーブル設計8を設計情報9としてファイルに一元管理し、この設計情報9をもとに、オブジェクト指向共通部品20として項目テーブル21、コードテーブル22、メッセージテーブル23を作成するとともに、FORMヘルパー24を構成し、メッセージ表示25、システムログ取得26を行う。この設計情報9はオブジェクト指向のプログラム製造工程及び項目テーブル21の作成に自動継承することができる。オブジェクト指向共通部品20は、項目テーブル、コードテーブル、メッセージテーブルに格納されたデータをカプセル化して製造工程の手続きに継承する。なお、設計情報（リポジトリ）9は設計ドキュメント28として出力してもよい。

【0026】モックアップ実行工程ではFORM29の機能を用いて画面に項目を打ち込み、FORMヘルパー24の機能を用いて、その画面入力の正当性を項目テーブル、コードテーブル、メッセージテーブルを参照して画面に表示することができる。ここで、FORMとはたとえばウィンドウズOS（米国マイクロソフト社の商標）で動作するコンピュータにビジュアルベーシックというソフトウェアをインストールしたときの画面定義体のことで、FORMという名の下に画面のみが現れ、その画面に向かって業務仕様を考慮して画面レイアウトや商品照会ルーチンならば処理区分や商品コードという項目を打ち込んでいけるものである。そして、FORMヘルパー24はこのFORMへのデータの入力および入力されたデータへの処理を助けるもので、例えば、画面への入力の桁数、属性が設計情報に照らして正当でなければエラー表示やエラーメッセージを表示したり、画面ヘコードの入力によってコードに対応する名称を出力したり、画面へ入力したコードが正しいか否かを画面に表示したり、画面の項目の入力をヘッド、ボディ、テイルの順に可能とする等の画面表示を行うものである。本発明は、このFORMヘルパーの動作により、画面に入力に対して、対応する出力を画面表示することにより、まだプログラムの設計段階であるのに、あたかも、プログラムが製造されているかのように顧客に対して見せること（これを本発明はモックアップ実行30と呼んでいる）が可能になる。更に、メッセージ表示25、システムログ取得26も各業務への共通ルーチンであるから、モックアップ実行工程で可能とする。

【0027】モックアップ実行30をモックアップ制御31の機能により行った後に、プログラムの製造工程2



7においては、自動継承された設計情報9をもとに各業務共通の業務スケルトンを業務スケルトン雛形32から選択コピーし、各項目の詳細なチェックである項目チェック33や各業務実行のためのファイル更新34の業務個別ルーチンを作成するとともに、ファイルを画面に転送するルーチンを自動生成35して業務スケルトン36、業務個別ルーチン、自動生成ルーチン35を用いてプログラムを製造する。そして、従来通りの実行レビュー36を行い、かつシステムテストを行う。

【0028】上記において、モックアップの実行モジュールはFORM29、モックアップ制御31、FORMヘルパー24を含むオブジェクト指向共通部品20からなる。業務アプリケーション実行モジュールはFORM29、オブジェクト指向共通部品20、製造工程作業27からなる。

【0029】本発明によれば、設計工程において画面に入力し、それに対応した出力を生じるモックアップ動作を実行できるので、プログラムの修正が早期かつ容易である。

【0030】また、設計工程で作成された項目情報をファイルとして保存し、製造工程にも継承できるので、プログラムの作成が効率的になる。モックアップ実行モジュールはFORMとモックアップ制御とオブジェクト指向共通部品とからなり、通常の業務アプリケーション実行モジュールは、FORMとオブジェクト指向共通部品と製造工程作業からなるので、FORMとオブジェクト指向共通部品が設計工程から製造工程に継承される。

【0031】従来はシステム要件に基づいて一度ワープロや紙等によって画面フォーマットの設計を行い、次に設計データを作成し、その後に画面上のFORMにシステム要件に基づく画面フォーマットを作成していたが、本発明はシステム要件に基づいて、いきなり画面FORMを作成するので、設計工程を簡略化するとともに設計情報の一元化が図れる。

【0032】

【発明の実施の形態】図2は全体フローチャートであり、特に設計工程を具体的に示す。先ず設計工程について説明する。

【0033】ターゲット言語、例えばビジュアルベシックのFORM機能を使ってFORMという画面定義体にレイアウトや項目を打ち込むことにより画面レイアウト41を作成する。次に、画面レイアウト41に使用された項目について項目ディクショナリ42を作成する。これは画面レイアウトに使用された項目のうち、項目ディクショナリ42にない項目については、その項目をいくつかの用語に分解し各用語の用語定義を行い、用語の組み合わせを求めて項目定義を行うものである。更に画面レイアウト41に用いられた項目について、各項目のうちその内容を更に具体的に規定すべきものについては、コード設計43としてコードの洗い出しを先ず行

い、コードと名称の対応関係を登録する。入出力情報定義(項目)44は例えば項目についてみると、項目ディクショナリから入出力情報項目、例えば各項目の項目名、項目ID、属性等を選択し、同時にFORMに各項目のIDを張りつける。これはFORMの各項目に対応してTAGを設け、そのTAGに各項目のIDを打ち込むことにより行う。

【0034】次にメッセージ、ボタン登録45を行う。メッセージの一元管理とボタンの一元管理のためにメッセージ、ボタン名称を登録する。そして、画面内グループ単位に初期表示メッセージ、ボタン名称を定義してアクションに関する入出力情報定義(アクション)46を行う。

【0035】設計工程で上述したごとく入出力情報定義体と画面表示との間に項目IDでリンクが張られているので、設計情報をもとにして画面処理を実行させ、設計の確認を行うことができる。本発明ではこの工程をモックアップ実行工程と称する。

【0036】製造工程においては、設計工程において生成された画面制御と入出力情報定義とのリンクをそのまま用いると共に、共通部品もそのまま用いてプログラムの再利用を図るとともに、ユーザごとの各種の要求に対応した業務固有ルーチンを作成するものである。

【0037】図3は本発明に係る全体システムの構成図であり、特に設計工程を具体的に示す。設計工程2においてはCASEツール50としては画面レイアウト、用語・項目管理、コード設計、入出力情報設計、メッセージ管理、ボタン管理を行う。図2に示した設計工程は、図3のCASEツールにより情報の一元化を図って行うことができる。

【0038】そして、設計の資産としては画面レイアウト3により得られたFORM51を有し、設計ドキュメントとして画面レイアウト書52を出力する。項目管理4は設計資産として用語ディクショナリ53と項目ディクショナリ54とを有する。コード設計6は設計資産としてコードテーブル55を有し、設計ドキュメントとしてコード設計書56を出力する。入出力情報設計5は設計資産として入出力情報57としての項目、アクション等を有し、設計ドキュメントとして入出力情報定義書58を出力する。なお、入出力情報57は入出力情報定義書58に加えて項目編集説明書59およびプログラム概要60、画面レイアウト書52も出力することができる。さらにメッセージ管理7およびボタン管理7は、それぞれ資産としてメッセージテーブル61およびボタンテーブル62とを有する。モックアップ実行工程30はモックアップツール63を用いて実行される。これについては後述する。製造工程27は上記設計工程2で作成された資産であるFORM51、用語ディクショナリ53、項目ディクショナリ54、コードテーブル55、入出力情報56、メッセージテーブル61、ボタンテーブ



ル 6 2、項目テーブル (図 4) を継承するとともに、各業務ルーチンに共通なたとえば画面制御等の共通部品資産 6 4 を使用してプログラムの再利用を図りながら行う。

【0039】ここで、図 2 及び図 3 で示された本発明のモックアップの開発手順を従来の開発手順と比較して説明する。通常の開発手順においては、設計し製造し終わった後で設計の段階に戻ったり、製造の開始時点に工程の戻りを行う必要があった。これは設計情報はドキュメントでしかなく、製造が終了する時点までイメージが湧かなかった為であり、更に実行していく仕様の変更もあるので、工程戻りの確率が極めて高かった。これに対して、本発明において図 1、図 2、及び図 3 に示すように、画面制御をモックアップとして実行し、プログラムすることなくデータエントリ等の画面処理をすることができるので、項目の入力チェック等の画面制御を実際に動く形式でユーザは確認することができる。したがって設計情報でモックアップが実行できるので、モックアップでの擬似実行に伴って設計工程を行うことになるので、通常の意味における製造工程終了に発生する工程戻りを減少させることができ、簡単に設計の追加変更を行うことができる。更に桁数チェック、必須入力チェック等の設計情報はモックアップで完了し、このモックアップで確認されたシステム開発作業における設計工程の情報 (FORM、項目定義等) を製造工程にも自動継承できるので、現実のプログラム製造工程ではモックアップに用いられたデータに関しては再利用可能である。

【0040】したがって本発明においては設計作業と製造作業と役割分担を明確にすることができ、これらを分業して行う場合等において設計情報の製造工程への自動継承を容易に行うことができる。

【0041】したがって、本発明によれば、モックアップ実行工程を経ることによりモックアップ実行によるメリットに加えて、そのモックアップ実行に用いられた資産がさらに製造工程に継承できるので製造工程も短縮できるというメリットもあわせてもつものである。

【0042】図 4 は本発明のシステム構成図であり、特にモックアップ実行及び業務アプリケーションを具体的に示す図である。設計工程 2 において画面レイアウト 3 はビジュアルベーシックの FORM 5 1 を用いて例えば処理区分等を画面上に表示する。コード設計 6 は処理区分の区分 (コード) 1、2、3 に対応してその名称を登録、修正、削除とし、処理区分の各区分をコードテーブル 5 5 に格納する。入出力情報設計 5 は例えば項目が処理区分ならば、項目 ID として SYRKBN を登録する。項目 ID とは項目をコンピュータが認識するためのものである。この入出力情報設計 5 により入出力情報から項目テーブル 7 0 を作成する。ボタン定義は B 0 0 0 には登録、取消、終了を定義し、それらは PF キー中の PF 1、PF 2、PF 3 に対応し、B 0 0 1 は取消のみ

定義し、PF 2 に対応する。メッセージ定義は E 0 0 0 としては“入力エラーです”として E 0 0 1 は“入力して下さい”とする。ボタン定義、メッセージ定義は、それぞれボタンテーブル 6 2、メッセージテーブル 6 1 に格納する。

【0043】次に、設計工程 2 からモックアップ実行工程 3 0 に移行するに際して、画面レイアウトはターゲット言語の FORM 機能で設計し、コード設計によってコードテーブルから例えば処理区分のコードを選択し、入出力情報定義によって項目テーブルを作成し、ボタン定義からボタンテーブル、そしてメッセージ定義からメッセージテーブルがそれぞれ作成される。

【0044】モックアップ実行工程においては、コードテーブル、項目テーブル、ボタンテーブル、メッセージテーブルからのデータを受けて FORM に対してモックアップ制御 7 1 および FORM ヘルパー 7 2 による制御が行われ、さらに共通ルーチンである業務共通部品 7 3 をアクセスする。モックアップ制御を行うためには、FORM ヘルパー内に設けられたチェックルーチンにより例えばデータエントリが登録されたデータの属性を満たしているか否かを判別し、入力に対して疑似出力を提供する。その際に表示画面の制御等の基本的なシステムとしてロギングやメッセージ表示等を業務共通部品 7 3 を用いて制御を行う。モックアップ実行 3 0 としては、たとえば上述の属性チェックの他に、桁数チェック、必須チェック等の (1) チェックルーチン、(2) コード参照入力及びコードチェック名称表示ルーチン、(3) グループ遷移、(4) メッセージボタン表示がある。

【0045】さらに固有業務アプリケーション 7 4 についてはモックアップ実行で用いられた処理を継承し、各業務に共通な業務スケルトン 7 5 以外の部分である、項目チェックやファイル更新ルーチンを作成する。FORM ヘルパー 7 2 や業務共通部品 7 3 はモックアップ実行で用いられたものを使用する。業務スケルトン 7 5 は再利用を行えるものであり、項目チェックは項目またがりや各項目の詳細内容をチェックし、ファイル更新は各業務の必要に応じてファイルを更新するものである。

【0046】図 5 は、設計工程において作成する FORM 一覧及び項目ディクショナリ、入出力情報定義との関係を示す。各画面には FORM ID が付され、それぞれ FORM 名称が付されている。まず、FORM 一覧を参照し、1 つの FORM を選択し、この選択された FORM について項目ディクショナリから入出力情報定義を作成する。

【0047】以下、図 2 の本発明のフローチャートを具体的に説明する。図 6 ～図 9 を参照して図 2 における項目ディクショナリの作成方法を具体的に説明する。

【0048】図 6 において用語ディクショナリ 5 3 には業務ルーチンに使用される日本語名および英数字名を入力し、登録する。図 7 において項目ディクショナリの管

理は用語組み合わせ処理において、用語ディクショナリから所定の用語を選択し、用語と用語の組み合わせで項目名及び項目IDを作成する。そして、項目名、項目ID、基本属性としての桁数、データ型等を項目ディクショナリ54に登録を行う。その際に各項目の基本属性として桁数、あるいはデータ型等を項目ディクショナリに入力する。

【0049】図8は用語ディクショナリ53の例である。用語と用語IDとが対応する。図9は項目ディクショナリ54の構成を示し、項目名、項目ID、桁数、データ型が対応する。桁数とは項目名に対応して入力される文字の桁数であり、またデータ型は文字型なのかあるいは数字型なのかを知らせるものである。なお桁数とデータ型は基本属性と称するものである。項目ディクショナリ54は用語ディクショナリ53から引いた用語の組み合わせでつくられる。例えば、処理区分SYRKBは処理SYRと区分KBとの組み合わせからつく。

【0050】図10及び図11はコード設計の作成方法およびコードテーブルを示す。図10において、FORMに打ち込まれた項目のうち、より具体的な設計を必要とするものに、コードIDを付してコードマスタ80

(図11)に登録する。コードマスタ80の1レコードには、コードID、コードID名称、属性、桁数、キー長、名称長が書き込まれる。例えば、コードID SYORIのコードID名称は処理区分である。次に、コードIDをテーブル名としてコードテーブル81、82を作成し、このコードテーブルにコードと名称を登録する。例えばコードID名称の処理区分のコードIDがSYORIのとき、SYORIをテーブル名としたテーブルのコードは1、2、3であり、それぞれの名称は照会、登録、更新である。

【0051】図12を参照して、図2における入出力情報定義〈項目〉の作成をより詳細に説明する。まず各FORMの1枚1枚に付された画面ID (FORM ID)を入力82し、次に項目ディクショナリ54から項目名とこれと同一レコードの項目ID、基本属性を選択する。画面定義体毎にFORMに使用する全項目を選択83する。そして、画面定義体 (FORM) 上ではその項目名に選択された項目IDをTAGに張りつける(84)。これと共に画面定義体に対応して入出力情報定義テーブル85の登録を行い、ここに選択された項目に対して1レコードを作成する。この入出力情報定義体の1レコードには画面ID、項目名、項目ID、基本属性、拡張属性をたとえば項目ディクショナリから選択して入力する。さらに、この1レコードに対して項目の拡張属性としてグループ番号 (画面項目のグループ化)、必須入力チェックをするかしないか、編集形式として、日付編集、金額編集、編集なし、あるいは項目のタイプとして入力項目、表示項目、ボタン項目等の入力を行ってもよい。画面定義体上の項目と入出力情報定義テーブルの

対応項目は同一の項目IDでリンクされる。

【0052】次に、たとえば処理区分という項目のように、コード1、2、3に対して登録、修正、削除というコード名称を参照する項目のため、コードIDを選択86するのであるが、そのコードIDはコードテーブル55に格納されている。そして、入出力情報定義テーブル85を更新し、入出力情報テーブル85のレコードに選択されたコードIDを登録87する。入出力情報定義テーブル85の1レコードはフォームID、項目名、項目ID、基本属性、拡張属性、コードID、手入力される名称表示項目からなることとなる。したがって、コードテーブルと入出力情報定義テーブル86とは同一のコードIDでリンクされることになり、入出力情報定義テーブル86からコードテーブル55の中身であるコード、名称を参照することができる。

【0053】図13は入出力情報定義テーブルを示す。このテーブルはたとえば画面 (FORM) ID、項目名、項目ID、桁数、データ型、グループID、必須、編集形式、項目タイプ、コードID、名称表示項目からなる。このうち画面ID、項目名、項目ID、桁数、データ型は項目ディクショナリにより検索することができるが、グループID、必須、編集形式は手入力情報であり、コードIDは、例えば処理区分についてSYRKUBUN、担当者コードについてはTANTOUをコードテーブル55から選択して入力する。名称表示項目は手入力情報であり、次に名称を表示すべき項目名、すなわち処理区分のときは処理区分名、項目IDSYRKBNMおよび担当者コードのときは担当者名の項目ID TANNMをそれぞれ手入力する。例えばグループIDは画面上の項目をヘッド、テイル、エンドとグループ化し、カーソルの動きを各グループ内に制限するものである。必須1の場合必ずチェックする必要がある、それ以外の場合は必ずしも必要ない場合がある。編集形式においては00は編集なしであり、10は通貨編集、例えば¥とかカンマを付ける等のものであり、20の日付編集は日付と月の間にスラッシュを入れるものである。項目タイプは入力項目、表示項目、およびボタン項目が存在する。

【0054】図14はメッセージテーブルを示し、例えばメッセージID E001のときは、『業務を終了して下さい』というメッセージをヘッド、ボディ、テイルのうちの所定のグループに対応して表示する。

【0055】図15はボタンテーブルを示す。例えばボタンID E001のメッセージについてボタン1は終了を示し、これはPFキーのPF1に対応する。図16は図2における入出力情報定義〈アクション〉46を示し、画面のグループ1 (ヘッド) について、メッセージID E001のメッセージとボタンID B001のボタン名称を表示することを示す。

【0056】図17は表示画面に対応するFORM51

およびその表示の制御を行うFORMヘルパー72、業務ルーチンの制御に共通して使用される共通ルーチンである業務共通部品73、そしてモックアップの制御71とからなる本発明が適用されるモックアップ制御システムのシステム構成図である。

【0057】画面にはFORM機能により処理区分、正方形、次に商品コード、長方形が打ち込まれる。FORMヘルパー72は項目テーブル(KTAG)70、FORMテーブル(FTAG)90、画面テーブル(G)91を参照しながら制御を実行する。項目テーブル(KTAG)は入出力情報定義テーブルから読み込んだ設計情報、FORMテーブル(FTAG)は項目テーブルをFORM単位に区分けするためのインデックス情報、画面テーブル(G)は項目テーブルと対になっている各FORM内項目アドレスをそれぞれ格納する。

【0058】次に、キー入力あるいはカーソルによるクリックによって各イベントが発生した場合のプロシジャを説明する。まず、イベントプロシジャの(1)として、Form-Lord イベントが到来した場合にはFORMヘルパーによりFORM、画面テーブル、項目テーブル、FORMテーブルの初期処理92が行われ、また業務共通部品73内において初期処理93されて、システムログ出力94を実行し、さらにモックアップ制御71中における業務スケルトン対応部において初期処理95が行われ、先頭グループをサーチするため次グループサーチ96を実行する。イベントプロシジャの(2)として、Form-Keypress イベントの場合には、FORMヘルパー72により桁数及び属性チェック97が行われる。イベントプロシジャの(3)としてForm-KeyDownイベントの場合にはFORMヘルパー72においてPFキーの受付98が行われ、イベントプロシジャ(4)の、Command-click イベントの場合にはFORMヘルパー72でコマンドボタン99を駆動する。FORMヘルパーのPFキー受付、コマンドボタンのいずれかが実行された場合にはモックアップ制御71の業務スケルトン対応部においてPFキー制御100が行われる。モックアップ制御においては、PFキー制御により入力完の時は次キーサーチ101に進み、取消の時は前グループを再度サーチ102する。PFキー制御100によりコード参照103する場合もある。さらに、PFキー制御100により、業務共通部品73を用いてメッセージ表示、ボタン表示103を行う場合もある。

【0059】イベントプロシジャ(5)としてカーソルが特定の項目に到来したときGot-Focusイベントが発生し、FORMヘルパー72において項目開始105が行われる。そして、項目において必要な入力がFORMに対して行われる。カーソルがその項目から離れるとイベントプロシジャ(6)でLost-Focus イベントの場合となりFORMヘルパー72において項目終了106が実行され、1項目の打ち込みを終了し、その1項目につい

て必須チェック107、項目編集108を起動し、更にこの場合、モックアップ制御の業務スケルトン対応部における項目チェック制御109も起動する。この項目チェックは各項目のより詳細な事項をチェックすることを可能にするものである。その後にコードチェック110が行われてもよい。なお、上記各イベント名について、例はビジュアルベーシックでのイベント名称であり、これはターゲット言語により違って来る。最後に業務共通部品73によって終了処理111する。

【0060】図18はモックアップ制御のフローチャートである。先ず、FORMヘルパー72が起動され、初期制御処理および項目テーブル(KTAG)及びFORMテーブル(FTAG)、画面テーブル(G)を作成し、次にモックアップ制御71に移り、初期処理を行って項目テーブルをサーチして先頭グループをサーチする。

【0061】次にFORMヘルパー72の制御に移り、Got-Focus イベントにより項目開始イベントを実行し、Form-Keypress イベントにより桁数、属性チェックを実行し、Lost-Focus イベントにより項目終了イベントが発生し、必須のチェックを行い、さらに項目編集を行ってもよい。これらをグループ内の項目数分だけ行う。

【0062】次にForm-KeyDownイベントを受けてPFキーを受け付け、或いはCommand-Click イベントを受けてコマンドボタンから入力する。その後制御は再びモックアップ制御71に移り、PFキー制御に従ってFORMヘルパーでのチェックエラーが残っていればグループ変更はせず、入力完PFキーの場合には次グループサーチし取り消しPFキーの場合は、前グループをサーチしてFORMヘルパーに依頼する。これにより見たいには指定したPFキーによりグループが遷移して見える。図19は、FORMヘルパー初期処理時のテーブル構造である。先ず、入出力情報定義テーブル85を入力し、これらを展開して項目テーブル(KTAG)70及びFORMテーブル(FTAG)90を作成する。項目テーブルは項目IDとグループID、その他属性よりなり、そのアドレスは0~項目数、例えば20までの配列番号である。項目IDには各項目のIDが記憶される。グループID及びその他属性はここでは省略する。FORMテーブル(FTAG)は画面毎にFORMIDを01、02、03として付し、各画面は項目テーブルのどのアドレスに対応するかを示す。例えば、FORMID02はポイント開10終14であるから、2番目の画面で項目テーブルの配列番号10~14に対応する。

【0063】前述したように、画面上の各項目に対してコントロールTAG120をペーストし、たとえば入出力情報定義テーブル85を画面に呼び出すことによりその項目の項目IDを検出し、そのコントロールTAG120にその項目IDを入れる。ここまで、設計工程で設計しておく。

【0064】すなわち、設計時にSEは画面のフォーム内項目アドレスに対応して設けられたコントロールTAG120に業務ルーチンに対応した項目IDを打ち込む。フォームヘルパーの初期処理としてこのコントロールTAG120の項目IDによって項目テーブル70の項目IDをマッチングして、合致した場合、同一項目IDの配列番号を前記コントロールTAG120の項目IDの次の領域99にたとえば10として書き込む121と共に項目の画面における位置に相当するフォーム内項目アドレスを画面テーブル122に書き込む123。画面テーブル122はFORMヘルパーの処理として、項目テーブル70の1つの配列番号に対応した1つの領域を設けることにより構成される。コントロールTAG120に項目テーブル70の1つの項目アドレスたとえば10が格納されると共にそのTAGがポイントする画面テーブル122の1つの領域にFORM内の項目アドレス3が格納されるので、コントロールTAG120によりFORM上の項目に対応する位置とその項目に対応する項目テーブルの1レコードとが関連づけられることになる。

【0065】図20は項目テーブル70を詳細に示す。項目テーブル70は図13に示した入出力情報定義テーブルから項目名を除いたものである。次に図21は、FORMヘルパーの初期処理をフローチャートで示す。このフローチャートはプログラム実行時に（モックアップを含む）最初の1回だけ呼ばれる処理で、項目テーブルと画面の各項目をリンク付けるものである。

【0066】最初に入出力情報テーブルを入力し、KTAG（項目テーブル）とFTAG（画面テーブル）を展開（ステップ130）し、それ以降はFORMヘルパーは項目テーブルの内容を配列変数、すなわち項目テーブルのアドレスとして参照できる。

【0067】次に画面の全項目を検査して、TAGに項目IDがセットされていれば、下記の2点のリンク付けを行う（ステップ131）。すなわち、画面FORMの各項目について配列番号IのTAGが未設定かどうかを、FORM内項目数、例えば0～20について検出し（ステップ132）、TAGが設定されていればすなわちタグに項目IDがセットされていれば、タグにセットされている項目IDとKTAGテーブルの全項目を比較し、等しい項目IDをサーチする。ヒットしたKTAGテーブルのアドレスをJに入れる（ステップ133）。次に、画面テーブルにFORM内項目アドレスをセットする（画面テーブル（J）=I）。その後TAGの後にKTAGインデックスを追加する。FORM（I）、TAG=FORM（I）、TAG+STR（J）（ステップ134）

上述したように、項目テーブルからFORMの所定領域に入力する場合には、項目テーブルのアドレスに対応する画面テーブルのアドレスを参照し、そのFORM内項

目アドレスに相当するところに項目テーブルの当該アドレスのデータを表示する。逆に、FORM内の特定の領域から項目テーブルを参照する場合には、FORM内項目アドレスが格納されている画面テーブルの特定のアドレスを検出し、TAGを用いて該アドレスに対応した項目テーブルのアドレスを参照してその項目テーブルの当該アドレスに記憶されている項目ID等を検出する。

【0068】このように項目テーブルから、画面上の位置をポイントすることも出来るし、あるいは画面上の特定の位置から項目テーブルの特定のデータをポイントすることもできる。これは、項目テーブルと画面テーブルとが相互にチェーンしているからである。

【0069】項目テーブルとFORMテーブルとはモックアップ時にFORMヘルパーの動作の一環として作成され、FORMやFORMヘルパーと共に、設計から製造へと継承されることになる。項目IDが入れられたTAGも各項目毎に作成され、設計から製造へと継承される。

【0070】画面テーブルは項目テーブルからFORM上のアドレスを指すように、あるいはその逆にFORM上の特定の領域から項目テーブルのデータを指すように、項目テーブルの特定の項目アドレスと画面上の項目のFORM内項目アドレスとを対応づけるものとして、FORMヘルパーの初期処理で各項目毎にその都度形成されるものである。またTAGへの項目テーブルの配列番号のセットも、FORMヘルパーの初期処理で各項目毎にその都度行われる。

【0071】図22はモックアップ内部仕様を詳細に説明する。先ず、画面レイアウトをビジュアルベーシックのFORM上に生成する。すなわち、画面にFORM機能によってFORM51に示すように、例えば商品照会である場合には、処理区分、縦長四角の図形、アンダーライン、商品コード、横長四角の図形、アンダーラインをSEが打ち込む。

【0072】次に、入出力情報定義テーブル85を画面を呼び出すと、その画面には、例えば項目名、項目ID、属性、コードID、表示名称が表示される。もし、FORMに打ち込むべき項目たとえば処理区分が入出力定義テーブル85に存在しないときは、更に項目ディクショナリ（図示せず）を表示し、その項目ディクショナリ内で処理区分を選択すると入出力情報定義テーブル85の画面に項目名として処理区分、IDとしてSYRKBN、および属性・・・が表示される。次にコードIDファイルを表示し、処理区分に対するコードIDを選択し、SYRKENを入出力情報定義テーブル85に表示する。さらに表示名称としては通常次の欄に記載されるべき項目名たとえば処理名のID、すなわちSYRNMを手入力すれば、次の項目の処理名が処理区分のコードを画面に入力したとき表示される。以下同様にしてシステム要件に対する業務ルーチンを作成するために処理

名、商品コード、商品名をFORMに入力する必要があるときにはこれを項目ディクショナリから選択して入出力情報定義テーブル85に入力表示する。

【0073】この後、モックアップ実行時に、入出力情報定義テーブル85に基づいてこれとほぼ同形の項目テーブル70を作成する。項目テーブルと画面の各項目に対して設けられた図示しないTAGとに基づいて、項目テーブル70と画面テーブル120との関連づけが行われる。なお入出力情報定義を直接設計ドキュメントとして出力し、プリントアウトすることも可能である。

【0074】項目テーブル作成ステップ137は入出力情報定義テーブル85の第1ラインのデータを項目テーブル70のアドレス0にSYRKBN, . . . , SYREBN, SHNNMとして登録し、さらに項目インデックス138を作成する。この項目インデックス138は製造工程で業務ルーチンが使用するものであり、項目テーブル70のテーブルアドレス0, 1, 2, 3をソースプログラムに書くのではなくて、このインデックスを書くことによって各アドレスに格納されている各項目情報の関連性と視認性をよくする。

【0075】次に、本発明の中心であるモックアップ処理について説明する。項目テーブル70のアドレス0には、項目IDSYRKBNの処理区分がストアされ、画面情報に対応する画面テーブル22のアドレス0には、画面上の項目ID SYRKBNのアドレス(2)が記憶される。したがって、画面テーブルにおいて項目テーブルとFORM画面との対応が形成される。したがって、まずSEが商品照会プログラムを設計するにあたって、FORM51に示す画面と必要な設計情報を作成したとする。その段階でユーザあるいはSEがFORM51を画面に表示して、処理区分の次の項目アドレスの長四角にコード1を入力するとFORMヘルパーの動作によってそのコードの桁数、属性等がチェックされエラーであるならその旨の表示が行われ、正しければコードテーブルを検索し、次のFORM内項目アドレス(3)に登録という名称が表示される。したがって、ユーザはまだプログラムが実際にできていないのに、コード入力1に対して何らかの出力がFORM上に現れるのでプログラムが動作しているかのように画面を見ることにより、その画面の動きに対して、さらなる希望や設計や設計変更を想到し、この変更をユーザからSEに伝達することができる。今は処理区分について説明したが、画面内の必要な項目情報を項目テーブルと関連させることができる。画面の各項目から項目テーブルのアドレスが特定もすることができるし、項目テーブルから画面の位置を特定することも出来る。

【0076】そしてテーブルアドレスをパラメータとした共通サブルーチン群が作成でき、属性チェック等が自動化し、モックアップが実現するものである。なお、各モックアップ処理は図示しないがチェックルーチンが実

行されることにより、データエントリの継承を行うことができる。

【0077】FORMヘルパーの動作として、桁数、属性チェック、コード参照コードチェックについて、以下、順次説明する。これらの動作によって、ユーザはSEが製造を実際に完了していない段階でプログラムが現実にも動いているような感覚で、すなわちプログラムのモックアップを用いて画面入力を行うことができる。

【0078】図23はモックアップ動作において、データエントリを使用するための自動属性チェック方式について説明する。画面にはFORMとして、処理区分、及び商品コードが表示されているとすると、イベントルーチンにおいては1文字エントリのチェック、すなわちキー入力チェックは属性チェック、桁数チェック、自動タブ(次のフィールドへの)が行われる。尚項目入力(ロストホーカスチェック)においては必須入力チェック、編集表示、コードチェックを伴う。項目テーブルは項目名として処理区分あるいは商品コードとそれに要求される属性とのテーブルである。属性としては、属性、桁数、自動タブ、必須入力、コードID、表示項目ID等をテーブルにしたものであり、この中で処理区分と商品コードは画面テーブルにFORM内項目アドレスとして(2)、(5)の対応関係が記憶される。

【0079】図24は、桁数属性チェックの動作フローチャートである。Got-Focus イベントは、項目開始イベントを駆動し、項目にカーソルが飛び込んだときFORM内項目アドレスをACTIVE-CTLに格納する(ステップ40)。次にキーボードから1文字を入力したために、Form-keypress イベントが生じた時、桁数チェックを実行する。その桁数チェックルーチンは先ずFORM(画面)内の現在カーソルがあるフィールドに基づいて項目テーブルのアドレス、すなわち配列番号K-IXをTAGのKTAGインデックスから取り出し、項目テーブルのK-IXアドレスから設計桁数Iを取り出し(ステップ141)、次に現在カーソルがあるフィールドのデータ長J(現在桁数J)を調べる(ステップ142)。そして両者の桁数を比較し、設計桁数Iが現在桁数Jより大の時にはそのまま処理が進行(ステップ143)し、NOの時には桁数オーバーのため現在カーソルフィールドの文字色をエラー色(赤)にする(ステップ144)。

【0080】次に属性チェックルーチンがForm-Keypress イベントによって駆動された場合には、項目テーブルKTAGから現在カーソルのあるフィールドの属性を取り出し(ステップ145)、その数字チェック(ステップ146)、英字チェック(ステップ147)あるいは日付チェック(ステップ148)を行い、エラー時にはBeep音をならしキーボードからの1文字をネグレストするようにする(ステップ149)。

【0081】図25は必須入力チェックのフローチャー

トである。Lost Focusイベントが発生すると項目終了イベントが駆動され、項目テーブルKTAGから今までカーソルがあった項目の必須桁数Iを取り出す(ステップ150)。そして、今までカーソルのあったフィールドのデータ長(入力桁数J)を調べる(ステップ151)。両者の桁数を比較する(ステップ152)。必須桁数Iが入力桁数Jと等しいかより小の時、必須入力項目に入力が打ち込まれたことを示すので処理が終了する。しかし、必須桁数Iが入力桁数Jより大の時には、必須入力の項目の桁数不足のため、今までカーソルのあったフィールドのバック色をエラー色(例えば赤)にする(ステップ153)。

【0082】図26は項目編集の例を示す説明図である。すなわち、入力値123456を入力した場合、編集IDが編集なし、数値編集、通貨編集であることに応じて、それぞれ編集後の表示フォーマットは123456、123, 456、¥123, 456となる。

【0083】19960101という数値が入力された場合には、編集IDが日付編集の時には1996/01/01と表示される。この仕掛けは項目毎にもっているタグの最後に編集前の値をセーブし、画面テキストエリアには編集後の値を表示する。すると、項目開始イベントのタイミングではセーブしたタグの内容を画面テキストエリアに表示すれば元の値になる。

【0084】図27は項目編集フローチャートである。Lost-Focusイベントが発生すると、項目終了イベントが駆動されて、必須入力チェックを行う(ステップ160)。次に項目編集を行うため項目テーブルKTAGから今までのカーソルがあった項目編集IDを取り出し(ステップ161)、この編集IDが0の時にはこのまま終了する(ステップ162)が、IDが存在する時には画面上のテキストエリアをタグの後にセーブし(ステップ163)、セーブした内容からテキストエリアに編集表示をし、数値編集、通貨編集、日付編集等を行う(ステップ164)。

【0085】図28は項目編集解除のフローチャートである。Got-Focus イベントが発生すると、項目開始イベントを駆動し、項目にカーソルが飛び込んだとき、そのFORM内項目アドレスをACTIVE-CTLにセーブし、ACTIVE-CTLではフォーム内項目アドレスをストアする(ステップ170)。

【0086】次に項目テーブルKTAGから現在カーソルがある項目の編集IDを取り出し(ステップ171)、その編集IDが存在しなければ処理が終了する(ステップ172)が、存在した場合には画面上のテキストエリアに、タグの後にセーブしてある編集前データをセットすることによって項目編集を解除する(ステップ173)。

【0087】本発明によると、SEがプログラムの設計段階でユーザに対して画面の処理区分の後に縦長四角に

処理区分のコードIDテーブルを見ながら、例えば“1”と入力した時のその桁数が正しければ、そのまま処理が進行し、誤っていればエラー色が表示されるので、実際にはプログラムが製造されていないのに、プログラムが出来上がっているような感覚で入力して画面に表示できる。したがって、ユーザがその時点でプログラムの仕様に変更等も提案できるのでよりユーザの希望に即したプログラムを設計できる。

【0088】図29は、コード参照及びコードチェックのフローチャートである。次に、実行時のFORMヘルパー処理を説明する。この処理には2通りあり、第1は検索PFキーが押圧され、コード参照処理を行う。この場合は、コードテーブルからコード情報を一覧して、選択入力させ結果を画面のコード及び名称エリアにセットする。

【0089】また第2はコードチェック処理を行う。この場合では、項目終了イベントは画面から入力されたコードを元にコードテーブルをサーチして有無を調べる。有ればそのコードの名称を名称エリアにセットする。

【0090】図30はFORMヘルパー処理のコード参照のフローチャートである。まず、Got-Focus イベントが入ると、項目開始イベント、すなわち項目にカーソルが飛び込んだ時、そのFORM内項目アドレスをACTIVE-CTLにセーブする(ステップ180)。

【0091】次に、Form-KeyDownイベントが入ると、PFキー受付し(ステップ181)、検索PFキーがNOの場合は終了し(ステップ182)、YESの場合は項目テーブル(KTAG)から現在カーソルがあるフィールドのコードIDを取り出す(ステップ183)。

【0092】 $K-IX = \text{FORM}(\text{ACTIVE-CTL}).\text{TAG} \langle \text{KTAGインデックス値} \rangle$

コードID=KTAG. コードID(K-IX)となる。次にコードIDをファイル名としてコードテーブルを入力し(ステップ184)、コード一覧表ウィンドウを表示する(ステップ185)。

【0093】次に、一覧表から1つ選択させる(マウス等使用)(ステップ186)。次に、選択された1行からコードと名称を取り出す(ステップ187)。選択コードは1行の先頭からキー長桁、選択名称は1行のキー長1桁から名称長桁である。

【0094】次に図31に示すように選択コードを画面の現在カーソルがあるフィールドにセットし、選択コードを画面に表示する(ステップ188)。

$\text{FORM}(\text{ACTIVE-CTL}).\text{Text} = \text{選択コード}$

次に、選択名称を画面の現在カーソルがあるフィールドの表示項目IDが示すフィールドにセットする(ステップ189)。

【0095】すなわち表示項目ID=KTAG. 表示項目ID(K-IX)

KTAG(項目テーブル)の項目IDを全項目にわたっ



てサーチする。KTAG、項目ID(1~全項目)=表示項目IDとなり、ヒットしたKTAGインデックスをIに入れる。

【0096】そして、KTAGインデックスを元に画面テーブルを使って選択名称をセットする。FORM(画面テーブル(I))、Text=選択名称である。

【0097】すなわち、選択名称に対応する項目テーブル(KTAG)のアドレスで画面テーブルを引き、そこに格納されているFORM内項目アドレスに選択名称を表示する。これにより、FORM上のコードの項目のたとえば次の項目にその名称を表示できる。

【0098】図32はFORMヘルパー処理のコードチェックのフローチャートである。まず、Lost-Focusイベントが入ると、項目終了イベントとなり、必須入力チェック及び項目編集となる(ステップ190)。次にコードチェックでは、KTAGから今までカーソルがあった項目のコードチェックフラグを取り出す(ステップ191)。

【0099】すなわち、K-IX=FORM(ACTIVE-CTL).TAG(KTAGインデックス値)

コードチェックフラグ=KTAG、コードチェックフラグ(K-IX)となる。次に、コードチェックフラグが0(チェック不要)の場合は終了し(ステップ192)、1(チェック要)の場合は、KTAGから今までカーソルがあった項目のコードIDを取り出す(ステップ192)。

【0100】コードID=KTAG、コードID(K-IX)となる。次に、コードIDをテーブル名としてコードテーブルを入力し(ステップ194)、次に図33に示すように今までカーソルがあった項目のテキストと比較する(ステップ196)。

【0101】等しいレコードが存在したか否か(ステップ196)で、存在しない場合には、入力されたコードデータが設計情報に見つからなかったため、エラーとみなし、今までカーソルがあったフィールドのバック色をエラー色(赤)にする(ステップ197)。

【0102】また、存在した場合には、ヒットレコードから名称を取り出す(ステップ198)。次に、ヒット名称を今までカーソルがあったフィールドの表示項目IDが示すフィールドにセットする(ステップ199)。すなわち、KTAG(項目テーブル)の項目IDをサーチする。

【0103】KTAG、項目ID(1~全項目)=表示項目IDとなり、ヒットしたKTAGインデックスをIに入れる。すなわち、KTAGインデックスを元に画面テーブルを使ってヒット名称をセットする。

【0104】FORM(画面テーブル(I))、TEXT=ヒット名称となる。すなわち、画面に入力したコードがコードテーブルに存在した時は、そのコード名称を表示項目IDが示すフィールドに表示する。

【0105】図34はグループ制御の説明図である。ヘ

ッドグループはボディ部で処理したいデータのキー入力を行い、ボディグループはヘッド部で指定されたデータの表示/修正を行い、テイルグループはボディ部で処理した結果の確認を行う。つまり、画面の動きはグループ単位のエントリとなり、順次グループ遷移していく。

【0106】図35はグループ制御の全体フローチャートである。まず、入出力情報定義として、画面内各項目を定義する時に、各項目にグループIDを指定するステップ200)。

【0107】次に、実行時の業務ルーチン又はモックアップ制御は実行処理の手順に従ってグループIDを決めてFORMヘルパーにグループスタート依頼する(ステップ210)。

【0108】次に、FORMヘルパーは、業務ルーチン(又はモックアップ制御)からグループスタート依頼を受けて、入出力定義からつくられた項目テーブルのグループIDを参照し、以降そのグループだけの入力を許す(ステップ202)。例えば、グループを0~9の10個まで指定できることとすると、アクティブグループの指定により、他のグループ項目のエントリを不可とする。

【0109】図36はFORMヘルパー処理のグループ制御のフローチャートである。まず、業務ルーチン(モックアップ制御)からグループスタート依頼を受けると、KTAG(項目テーブル)を元に全項目に対し、TABSTOPプロパティを指定する(ステップ210)。すなわち、KTAG、グループID(1)=ACTIVEグループであり、YESの場合 → TABSTOP=ONとなり、NOの場合 → TABSTOP=OFFとなり、IをFORM内全項に対して行う。

【0110】以降TABキー又はリターンキーを押した時、TABSTOP=ONの項目にしかカーソルが飛ばない。次に指定されたグループIDに対応するメッセージIDあるいはボタンIDを元にそれぞれサブルーチンを呼び出す(ステップ211)。

【0111】図37において、メッセージボタン表示の実行を説明する。まず、FORMヘルパーが初期処理として入出力情報定義体のアクションテーブル(図16)をFORMヘルパーが初期処理として展開し、次にモックアップ制御または業務ルーチンがグループIDを指定してスタート依頼を行う。次にFORMヘルパーは指定したグループIDに対応するメッセージIDをもとにメッセージ表示のサブルーチンを呼び出し、或いは指定されたグループIDに対応したボタンIDをもとにボタン表示サブルーチンを呼び出す。すなわち、FORM IDがABC12301の画面は、そのグループ1について、メッセージE001がそのグループ7においてメッセージE0015が表示され、図4には図示しないが、図16よりグループ1ではボタンB001がグループ7ではボタンB003が駆動される。



【0112】図38は部品資産継承の動作のフローチャートである。先ず、フォームの作成に際して、(1) ひな型フォームをコピーしてターゲットフォーム名を変名して、新しいFORM ID をつくる。次に、(2) 画面レイアウトを作成する。入出力定義画面を呼び出して、(3) 入出力定義作業として各項目を定義する。そして(4) 入出力情報定義ファイルあるいは及び項目インデックス転送ルーチン等の自動生成ルーチンへと転送する。

【0113】次に、モックアップ制御が行われて、(5) ひな型モジュール構成をコピーしてターゲット名に変名する。そして、(6) フォームモジュールすなわち(2) でつくったFORM ID を追加し、(7) モックアップ実行し、設計情報のデバッグおよび業務仕様レビューを行う。(1) ~ (7) が設計工程作業である。

【0114】次に製造工程作業では、(8) ひな型スケルトンから業務スケルトンをコピーし変名する。そして(9) モジュール構成を変更する。モックアップを制御を解放し、変名した業務 スケルトンを追加し、自動生成ルーチンを追加する。

【0115】次に(10)業務スケルトン修正作業を行い、グループ制御部分の見直しと項目チェック、関連チェックルーチンを追加し、ファイル入出力ルーチンを追加し、最後に(11)実行テストを行い開発を終了する。

【0116】図39は部品資産継承図である。先ずひな型ファイルからひな型を(1) copyして、FORMファイルに入力し、FORMとして(2) 画面レイアウトを作成する。なお、ステップ(1) ~ (11)は前述の図38のステップ(1) ~ (11)に対応する。

【0117】次に、(3) 入出力情報定義を作成し、これに基づいて入出力情報定義と、自動生成(4) ルーチンとを作成する。次にひな型ファイルからひな型を(5) コピーしてモジュール構成(MAK)とし、これにはモックアップ制御、FORMヘルパー、業務共通部品が含まれる。モックアップ時にはモックアップ制御、FORMヘルパー、業務共通部品に加え、(6) FORMモジュールを追加してからこれらを(7) 実行する。

【0118】さらに製造工程作業としてエントリをひな型ファイルから(8) コピーして業務ルーチンファイルに格納する。そして、モックアップ制御を(9) 解放し、FORMヘルパーと業務共通部品はそのまま継承し、(8) でコピーした業務スケルトンと入出力情報定義から自動生成されたルーチンを追加し、(10)修正して継承し(11) 実行する。ここで、モジュール構成ファイルとはプログラム実行単体に存在し、当該プログラムが持つモジュールのファイル名をテーブルとして持っている。修正事項はこのモジュール構成ファイル単位に行うことができる。例えば、前述の項目テーブル(KTAG)、フォームテーブル(FTAG)はそれらが作成される都度製造工程に継承される。

【0119】図40は本発明のFORMヘルパーとモッ

クアップ制御を継承して製造されたプログラム全体システム構成図である。この図は図17に示したモックアップ制御のシステム構成図と、業務ルーチン220を除いて同様である。すなわち、全体システム構成図は業務ルーチンの実行時を示すのに対して、モックアップシステム構成図はモックアップ制御時を示す。全体システム制御において各業務に共通な業務スケルトンとして、初期処理95及びPFキー制御100、及び項目チェック制御109はモックアップ制御と同様であるが、業務固有ルーチン、自動生成ルーチンが業務ルーチンとして加わった。

【0120】業務ルーチンにおいては業務スケルトンにおいてPFキー制御が行われた後、先ずヘッド関連チェック221が行われ、NGの場合は共通ルーチンのメッセージ表示あるいはボタン表示を駆動する。OKの場合には各業務ルーチン用のファイル入力222し、その後画面にデータの項目転送223を行いグループをボディに移す。

【0121】次に、ボディにおいてボディ関連チェック224を行い、NGの場合にはやはり共通ルーチンにおいてメッセージ表示、ボタン表示を行う。グループティールの場合には、業務ルーチンにおいて画面よりデータを項目転送226し、ファイル更新を行う。業務ルーチンのうち、ヘッド関連チェック、ファイル入力、ファイル更新、項目チェックは業務固有ルーチンであり、項目転送は自動生成ルーチンである。

【0122】図41は本発明によるモックアップ動作の画面を示す。

- ① メッセージテーブルよりメッセージの表示
- ② ボタンテーブルよりボタンの表示
- ③ 項目テーブルよりFORM内の項目のグループを化を行う。これによりグループ以外の項目には、カーソルが行かなくなる。

【0123】④ 検索実行(PFキー、又はボタン等)(検索ウィンドウ表示)

- ⑤ コードテーブルからコード及び名称のデータ表示
- ⑥ 選択データをFORM内の処理区分の次の項目アドレス位置(長四角内)にコードのチェック結果をその次のアンダーライン部に名称をそれぞれ反映

⑦-1 入力フィールドチェック

【0124】

- |   |   |                    |
|---|---|--------------------|
| <ul style="list-style-type: none"> <li>・属性</li> <li>・桁数</li> <li>・必須入力</li> </ul> | } | チェックエラー時はエラー色(赤)表示 |
|---|---|--------------------|

【0125】⑦-2 編集

- ・数字 ¥、カンマ、等
- ・日付 年月日、時間等

⑦-3 商品コードを入力。

## 【0126】⑧ エンターポイント処理

グループの最終にエンターポイントを定義。このボタンを押すことで、グループ内の処理を完了し次のグループへ移る。

【0127】図41に示した画面上でユーザまたはSEは項目等の入力データの正当性のチェックコードを入力することによるコードや名称の参照、コードの正しいか否かのチェックあるいは入力データの編集を各グループ毎に遷移して行うことができるので、設計段階であるにも係わらずFORMが入出力関係に従って変化するので、実際にプログラムが製造された場合の画面の変化を実現しかつ検証できる。

## 【0128】

【発明の効果】以上詳述したように本発明によれば、プログラムシステムの設計から製造にいたる過程において、製造工程が終わるのを待って手戻りをして、設計工程あるいは製造工程を直すという従来のやり方を改善し、設計情報からモックアップを活用して少なくともデータエントリに関しては、実際にデータエントリの業務プログラムが動作しなくても結果として表示画面にはデータエントリが行われたか否かを示すことができるようにしたソフトウェア開発ツールを提供するものである。

【0129】このように作られた設計情報はそのままコード情報として製造工程に自動継承することができるので、製造工程におけるプログラムを行う必要がない点で極めてプログラムの作成工程を簡略化できる。

【0130】モックアップ動作の実行にあたってはフォームヘルパー内に設けられているチェックルーチン、共通部品内のフォームに設けられたフォームヘルパー用の共通ルーチン或いは業務共通部品とをそれぞれ用いてデータエントリがあくまでも行われたかの如くみせることができる。

【0131】またフォームヘルパーの中のチェックルーチン、共通部品の中の共通ルーチンあるいは業務共通ルーチンを再利用することができるという大きな効果も存在する。

【0132】さらに、紙やワープロで画面レイアウトを行うのではなく、FORMに直接画面レイアウトを行い、入出力情報等をCASEツール入力を行うので、設計データの一元化も図れる。

## 【図面の簡単な説明】

【図1】本発明の原理説明図である。

【図2】本発明の全体フローチャートと特に設計工程を示す図である。

【図3】本発明のシステム構成と特に設計工程を示す図である。

【図4】本発明のシステム構成と特にモックアップ構成を示す図である。

【図5】FORMと入出力情報定義と項目ディクショナリの関係を示す図である。

【図6】用語ディクショナリ作成方法を示す図である。

【図7】項目ディクショナリの作成方法を示す図である。

【図8】用語ディクショナリを示す図である。

【図9】項目ディクショナリを示す図である。

【図10】コード設計、コードテーブル作成のフローチャートである。

【図11】コードテーブルの構成図である。

【図12】入出力情報定義作成のフローチャートである。

【図13】入出力情報定義テーブルを示す図である。

【図14】メッセージテーブルを示す図である。

【図15】ボタンテーブルを示す図である。

【図16】入出力情報定義のアクションを示すテーブルの図である。

【図17】モックアップの概略システム構造を示す図である。

【図18】モックアップ動作のフローチャートを示す図である。

【図19】FORMヘルパー初期処理時のテーブル構成図である。

【図20】項目テーブルを示す図である。

【図21】FORMヘルパーの初期処理時のフローチャートである。

【図22】FORMヘルパーの処理を示すテーブル関係図である。

【図23】属性チェックの説明図である。

【図24】桁数・属性チェックのフローチャートを示す図である。

【図25】必須入力チェックのフローチャートを示す図である。

【図26】編集例を示す図である。

【図27】項目編集のフローチャートを示す図である。

【図28】項目編集解除のフローチャートを示す図である。

【図29】コード参照、コードチェックの全体フローチャートを示す図である。

【図30】FORMヘルパーのコード参照フローチャート（その1）である。

【図31】FORMヘルパーのコード参照フローチャート（その2）である。

【図32】FORMヘルパーのコードチェックのフローチャート（その1）である。

【図33】FORMヘルパーのコードチェックのフローチャート（その1）である。

【図34】グループの説明図である。

【図35】グループ制御全体フローチャートである。

【図36】FORMヘルパーのグループ制御フローチャートである。

【図37】入出力情報定義のアクション動作の説明図で

ある。

【図38】部品資産継承のフローチャートである。

【図39】部品継承を説明するためのブロック図である。

【図40】FORM、FORMヘルパー業務共通部品を継承して製造した業務ルーチンアプリケーションの構成図である。

【図41】本発明によるモックアップ動作およびこれを継承した業務ルーチンアプリケーション動作を表示するディスプレイ図である。

【符号の説明】

- 1 モックアップ実行レビュー
- 2 設計工程
- 3 画面レイアウト
- 4 项目管理情報
- 5 入出力情報設計
- 6 コード設計
- 7 メッセージ、ボタン管理情報
- 8 テーブル設計

- 9 設計情報
- 20 オブジェクト指向共通部品
- 21 項目テーブル
- 22 コードテーブル
- 23 メッセージテーブル
- 24 FORMヘルパー
- 25 メッセージ表示
- 26 システムログ取得
- 27 製造工程
- 28 設計ドキュメント
- 29 FORM
- 30 モックアップ実行
- 31 モックアップ制御
- 32 業務スケルトン雛形
- 33 項目チェック
- 34 ファイル更新
- 35 自動生成ルーチン
- 36 業務スケルトン

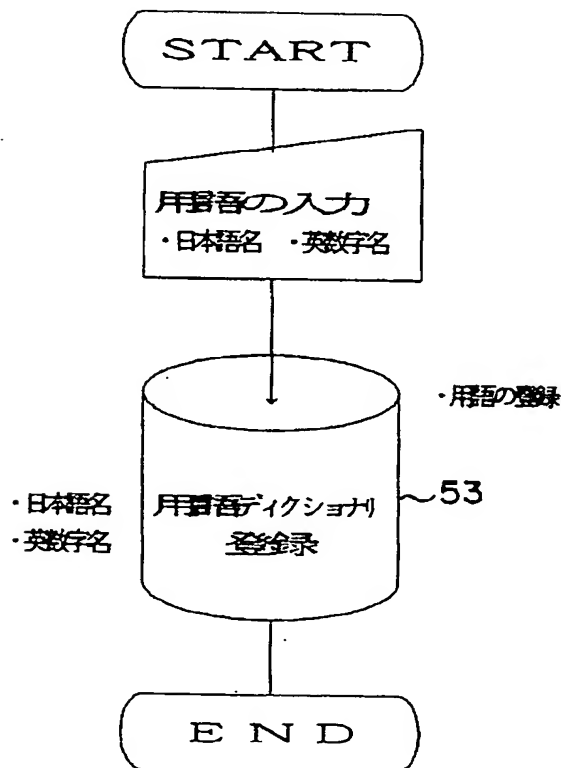
【図6】

【図8】

【図14】

# 用語ディクショナリ作成方法を示す図用語ディクショナリ

メッセージテーブル

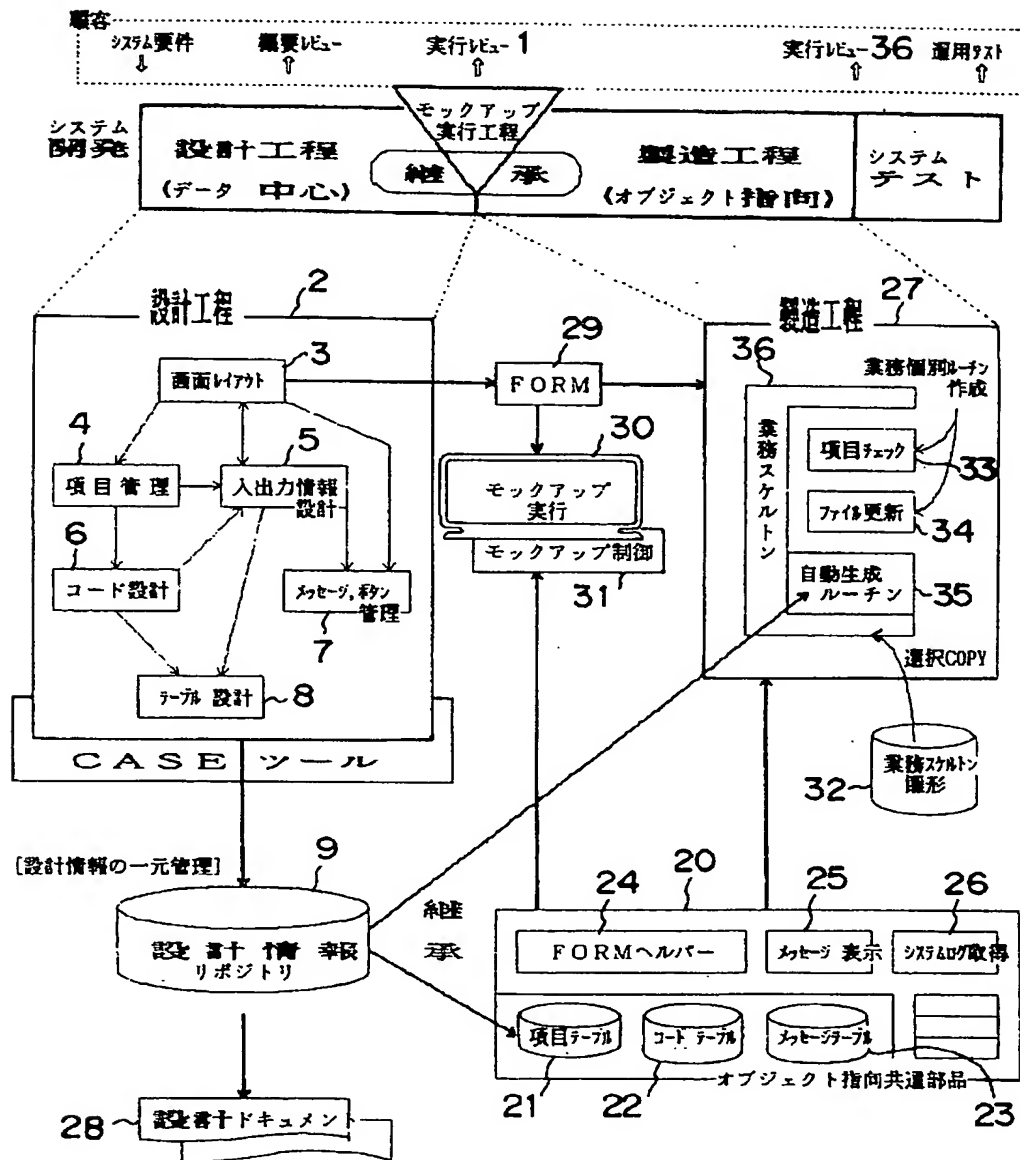


用語名	用語ID
処理	SYR
商品	SYO
契約	KYK
出荷	SYU
担当者	TAN
名	NM
区分	KB
金額	KIN
コード	CD
日付	YMD
印刷実行	PRT
ボタン	BTN

メッセージID	メッセージ
E001	登録を終了して下さい。
E002	%1以上%2以下の値を入力して下さい。

【図1】

## 本発明の原理説明図



【図26】

編集例を示す図

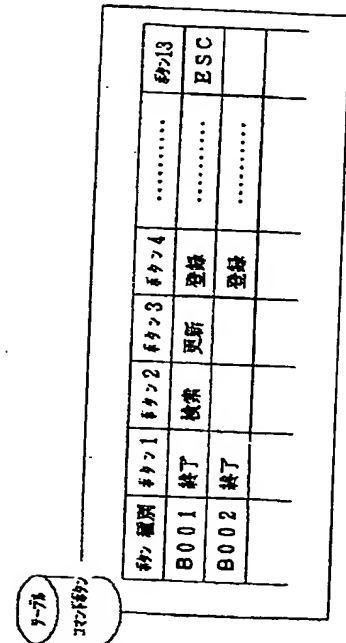
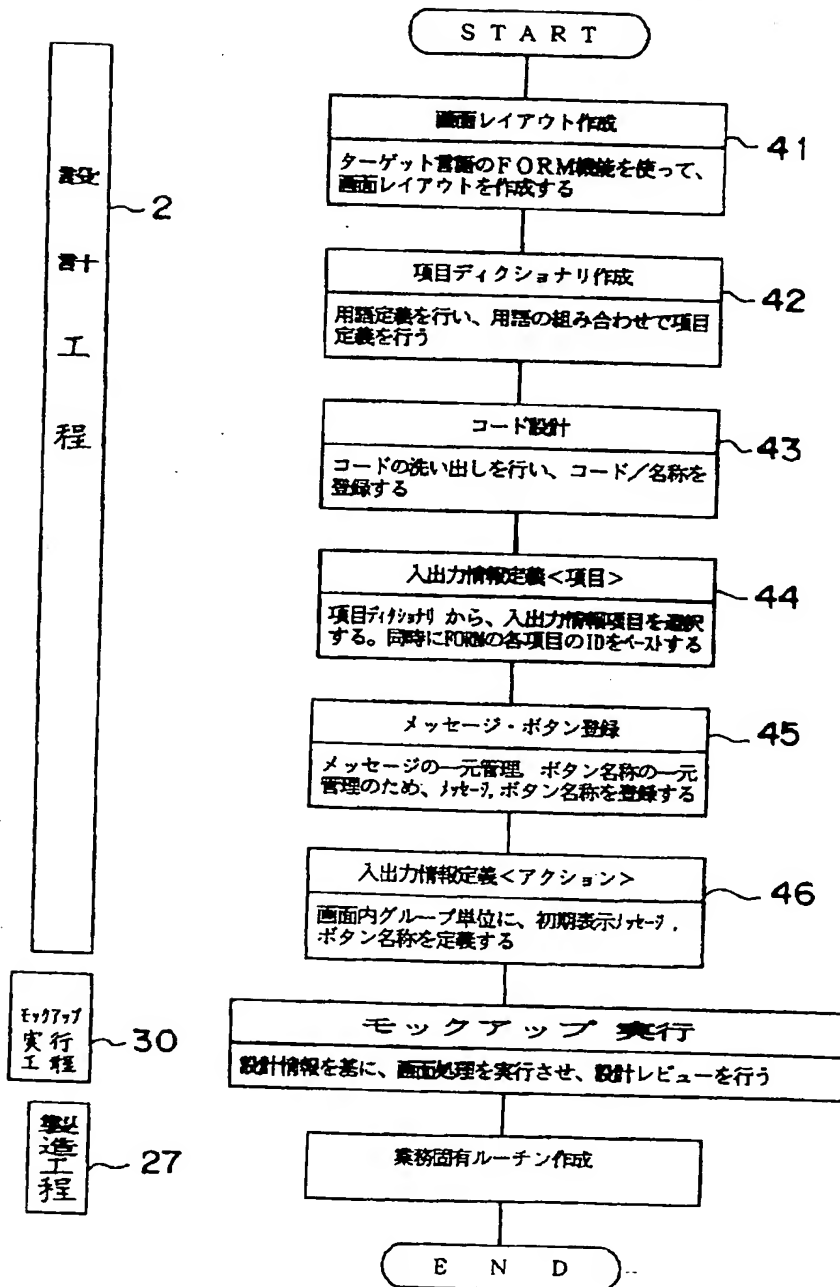
入 力 値	編集ID (説明)	編集後の表示フォーマット
123456	00 (編集なし)	123456
	22 (数値編集)	123,456
	32 (通貨編集)	¥123,456
19960101	52 (日付編集)	1996/01/01

【図2】

【図15】

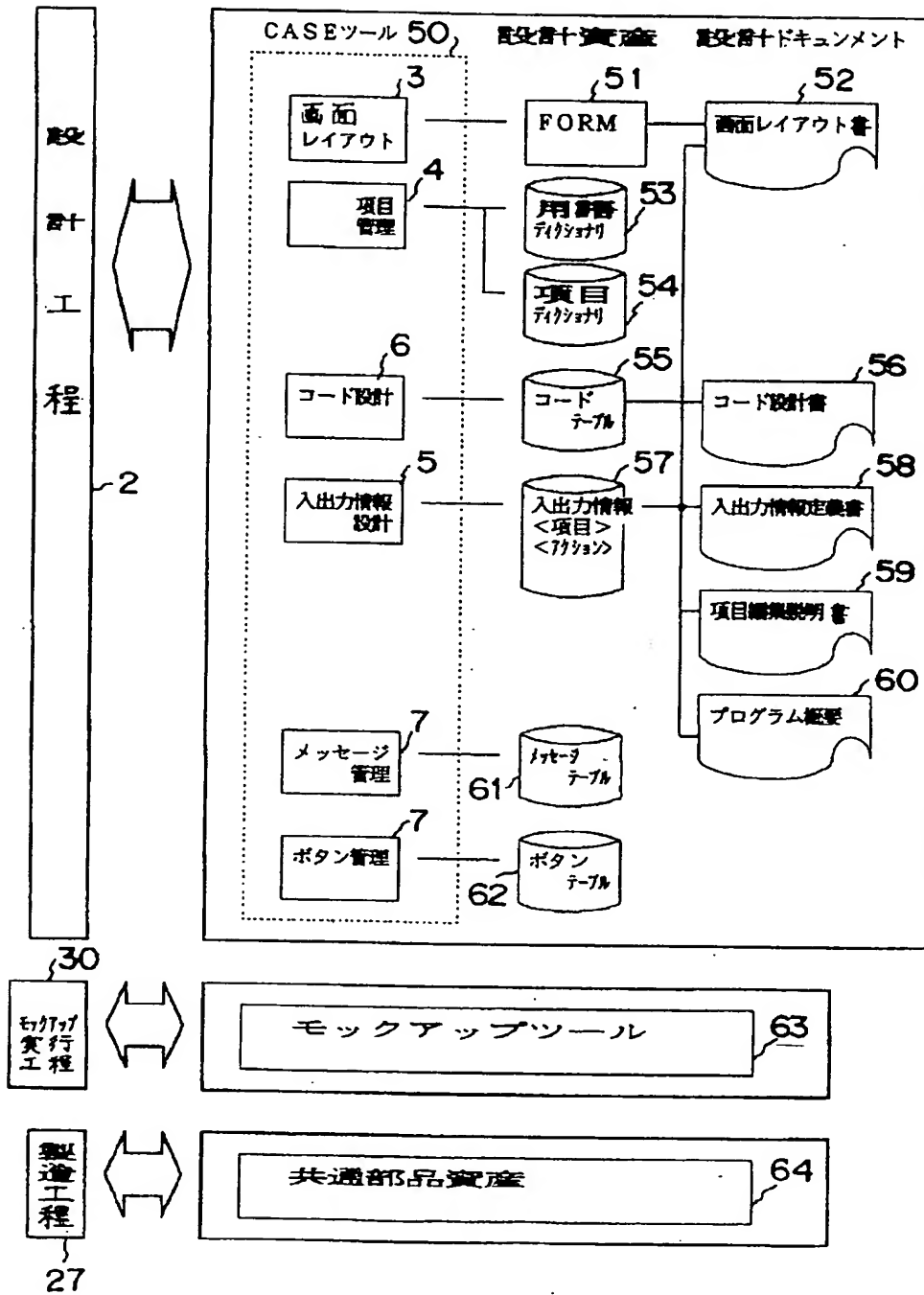
本発明の全体フローチャートとくに設計工程を示す図

ボタンテーブル

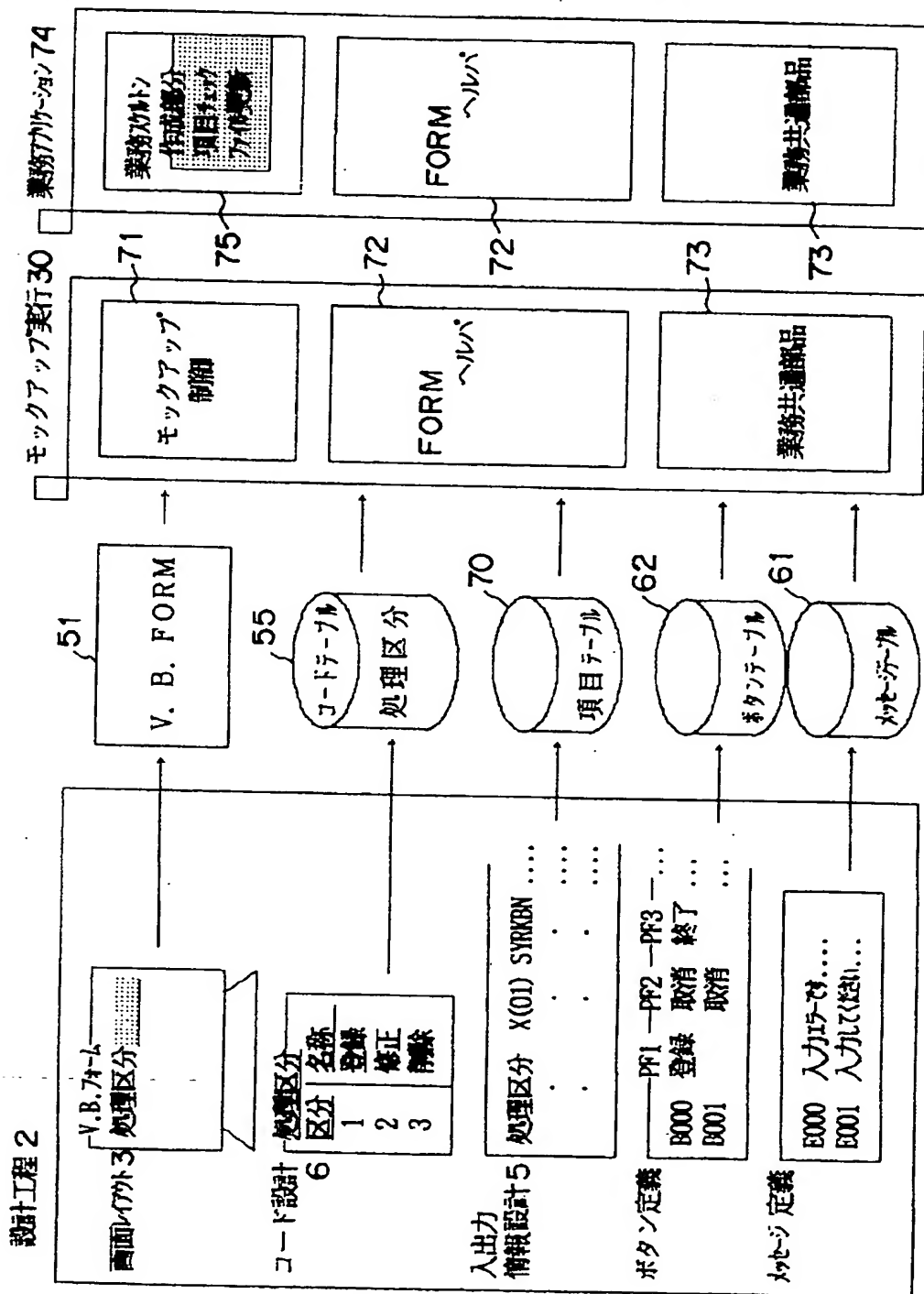


【図3】

本発明のシステム構成とくに設計工程を示す図



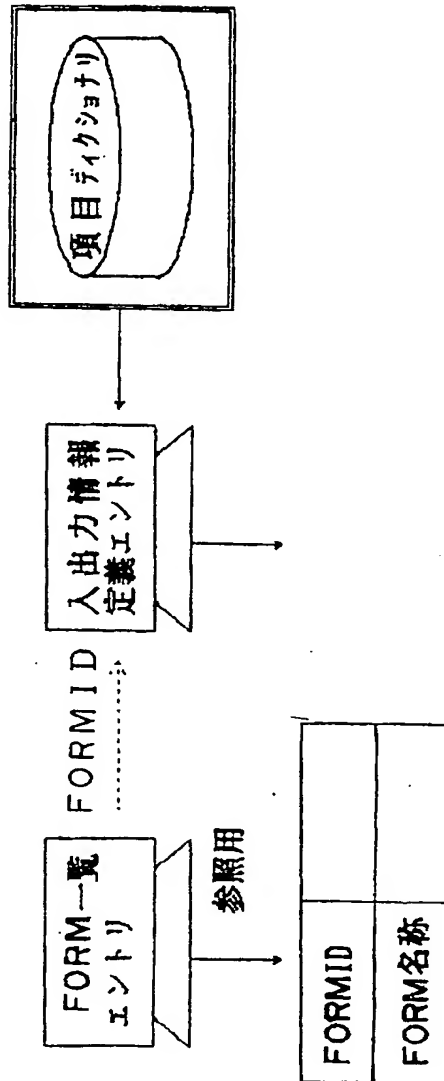
本発明のシステム構成ととくにモックアップ構成を示す図





【図5】

FORM と入出力情報定義と  
項目ディクショナリの関係を示す図



【図9】

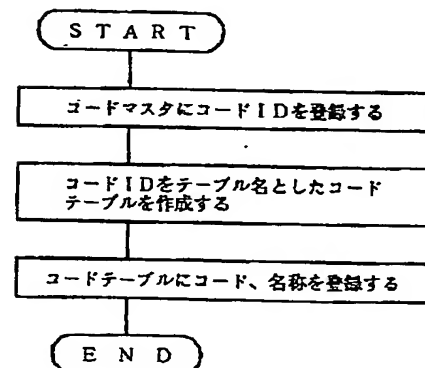
項目ディクショナリ

項目名	項目ID	桁数	データ
処理区分	SYRKB	1	文字型
処理区分名	SYRKBNM	10	文字型
商品コード	SYCOD	4	文字型
商品名	SYCNM	20	文字型
商品区分	SYCKB	1	数値型
商品金額	SYCKIN	8	数値型
処理区分	SYRKB	1	文字
契約区分	KYKKB	1	文字
契約名	KYKNM	30	文字
契約日付	KYKYMD	8	数値型
契約金額	KYKIN	8	数値型
担当者コード	TANCD	3	文字型
担当者名	TANNM	15	文字型
出荷日付	SYKYMD	8	数値型
印刷実行秒	PKTBTM		

基本属性

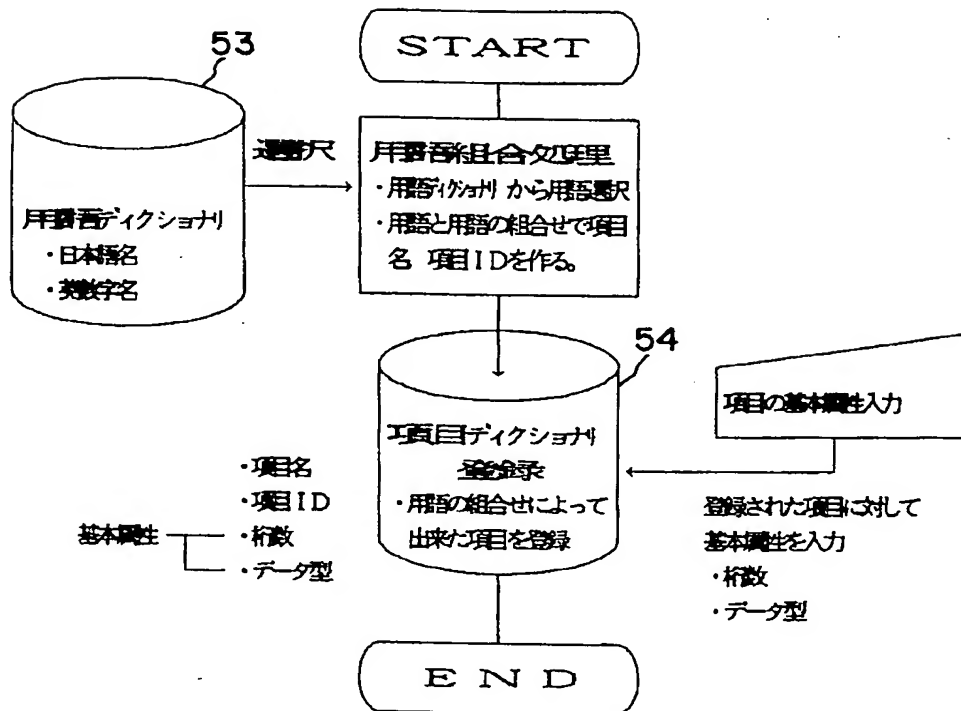
【図10】

コード設計、コードテーブル作成のフローチャート



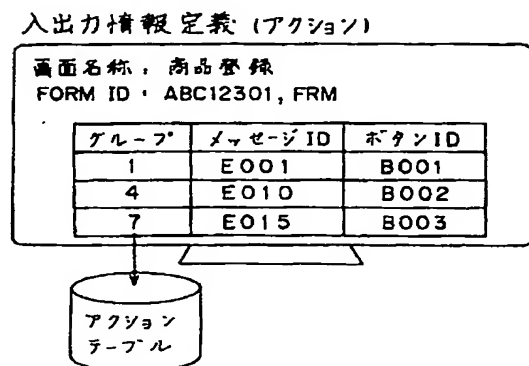
【図7】

項目ディクショナリの作成方法を示す図



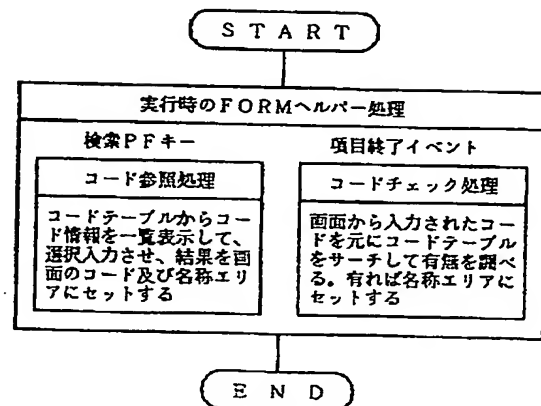
【図16】

入出力情報定義のアクションを示すテーブル



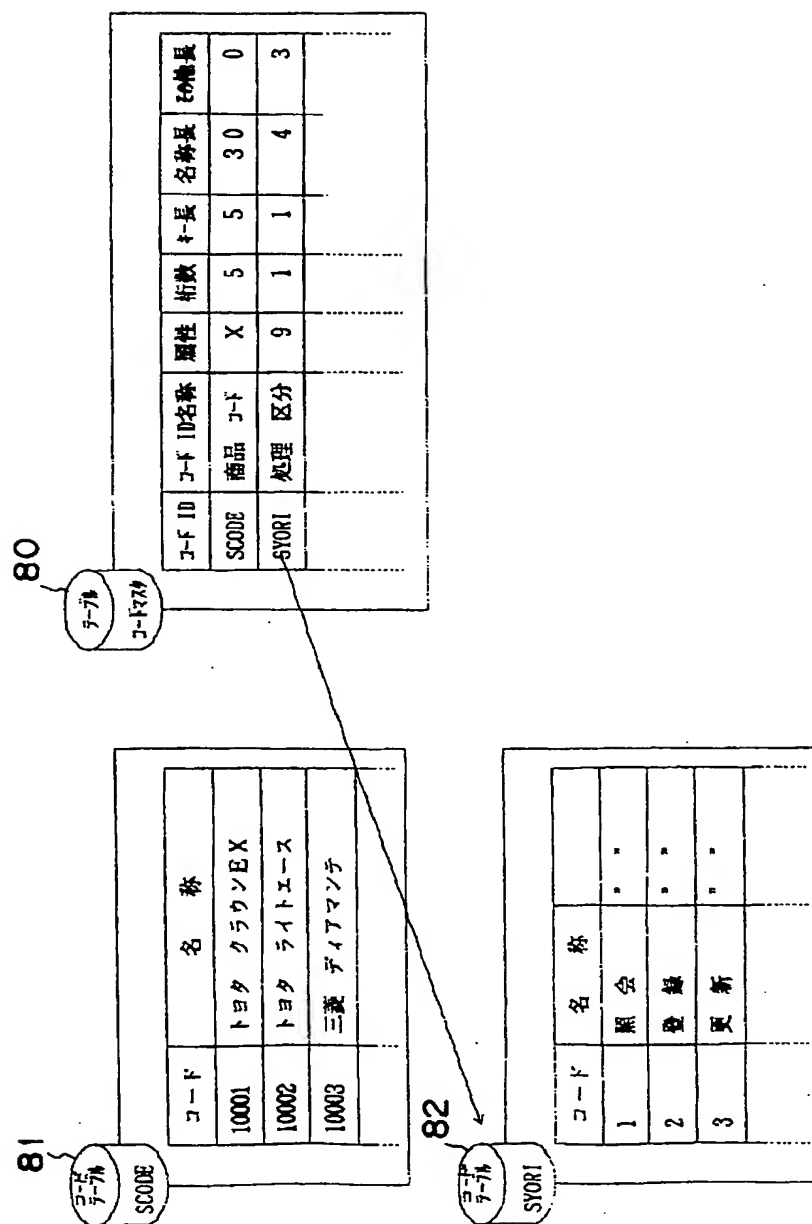
【図29】

コード参照、コードチェックの全体フローチャート



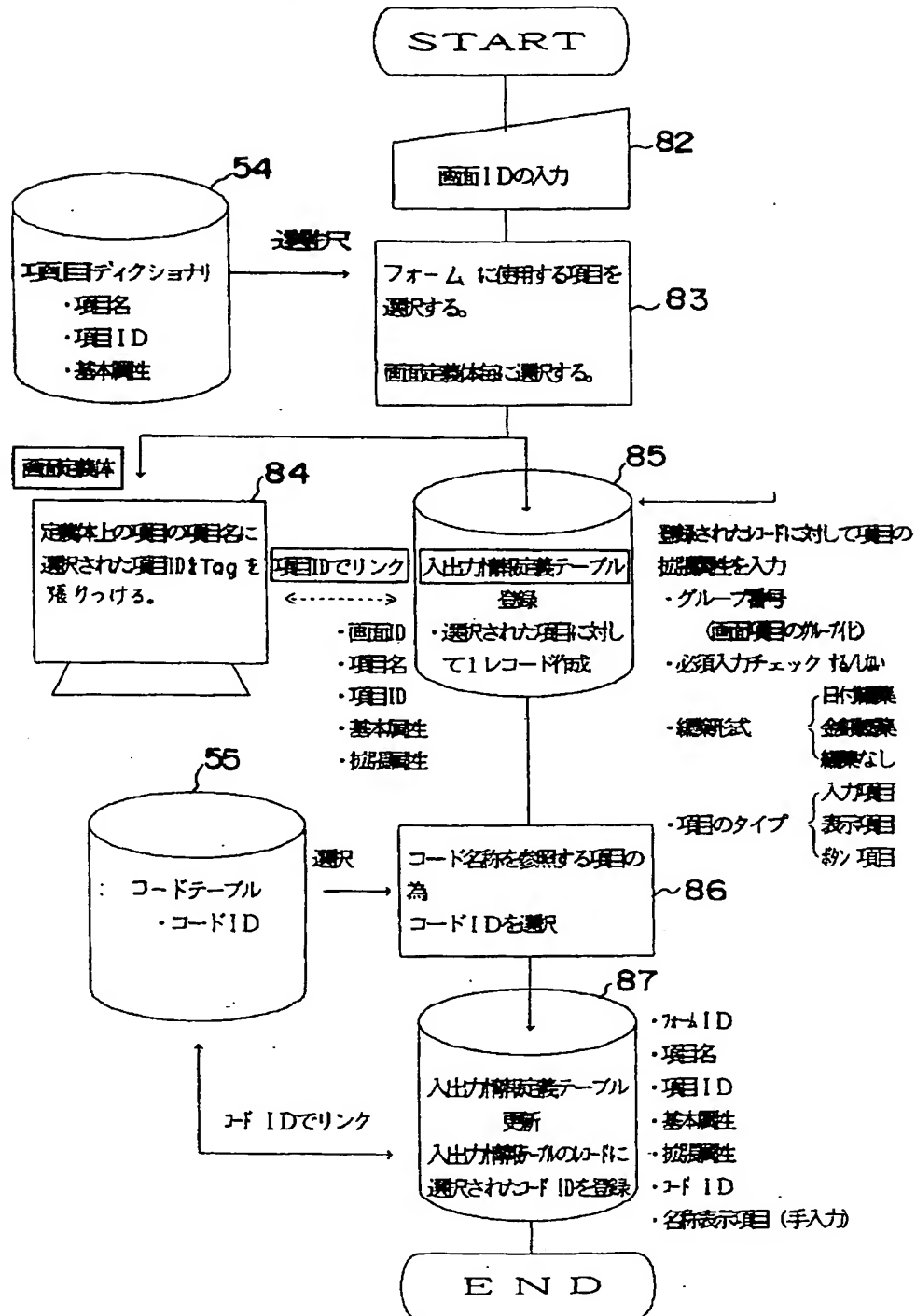
【図 11】

## コードテーブルの構成図



【図12】

## 入出力情報定義作成のフローチャート



【図13】

## 入出力情報定義テーブル

画面ID	項目名	項目ID	桁数	データ型	キーID	必須	編集形式	項目タイプ	コードID	名称表示項目
SY0ENTRY	処理区分	SYKCB	1	文字型	1	1	00	TXT	SYKCBUN	SYKCBNM
SY0ENTRY	処理区分名	SYKCBNM	10	文字型	1	0	00	LBL		
SY0ENTRY	商品コード	SYOCD	4	文字型	2	1	00	TXT		
SY0ENTRY	商品名	SYONM	20	文字型	2	1	00	LBL		
SY0ENTRY	商品区分	SYOKB	1	数値型	2	0	00	TXT		
SY0ENTRY	商品金額	SYOKIN	8	数値型	2	1	10	LBL		
KYKINPUT	契約区分	KYKCB	1	文字	1	1	00	TXT		
KYKINPUT	契約名	KYKCNM	30	文字	1	1	00	LBL		
KYKINPUT	担当者コード	TANCJ	3	文字型	1	1	20	TXT	TANTOU	TANNM
KYKINPUT	担当者名	TANNM	15	文字型	1	0	10	LBL		
KYKINPUT	契約日付	KYKYMD	8	数値型	2	0	20	TXT		
KYKINPUT	契約金額	KYKIN	8	数値型	2	1	10	TXT		
KYKINPUT	出荷日付	SYKYMD	8	数値型	2	0	20	LBL		
KYKINPUT	印刷実行ボタン	PRYBTM			3			BTN		

項目ディクショナリから選択

手入力情報

コード管理  
手入力情報  
キーIDから選択

【図34】

グループの説明図

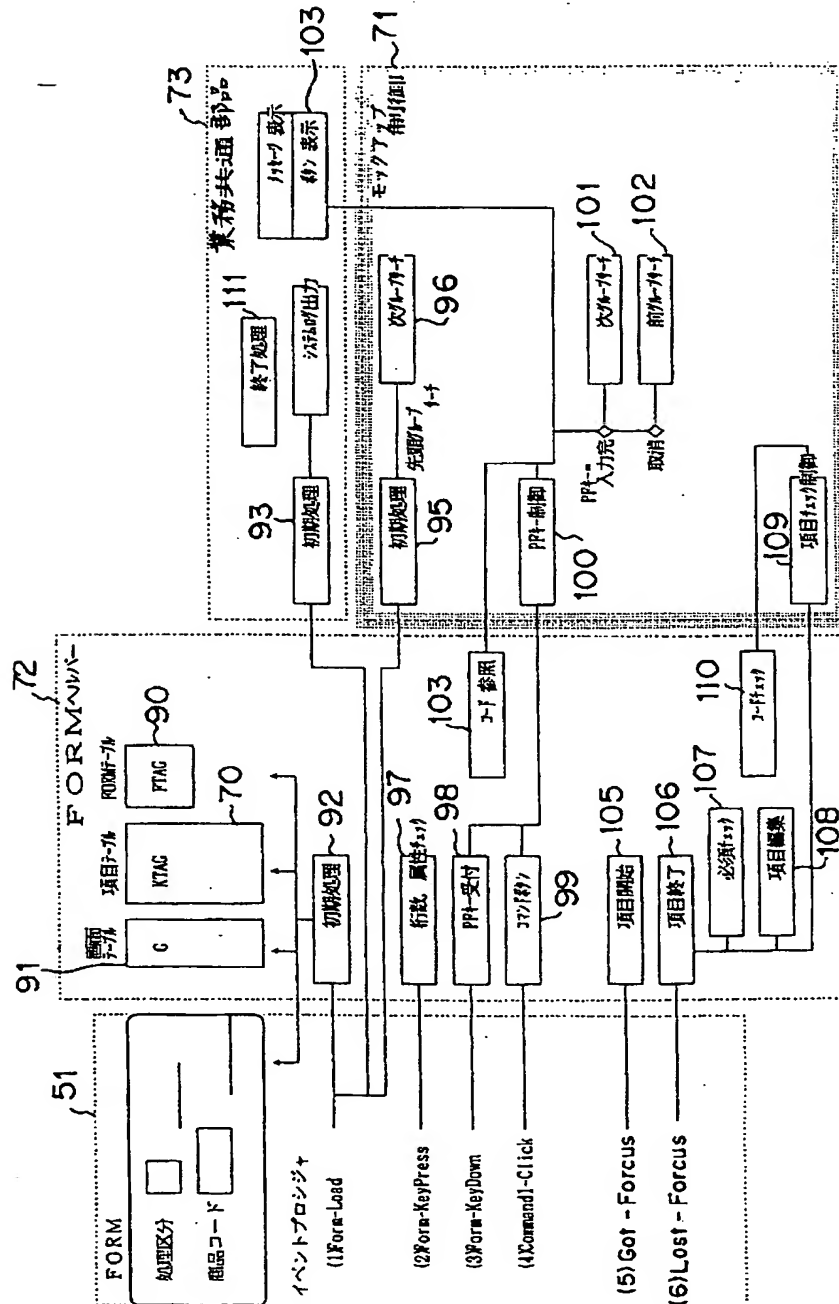
ヘッダグループ { 処理区分 ☐ 商品コード ☐

ボディグループ { 商品色:   
イメージ 大きさ:   
重さ:

フィルグループ { ☐

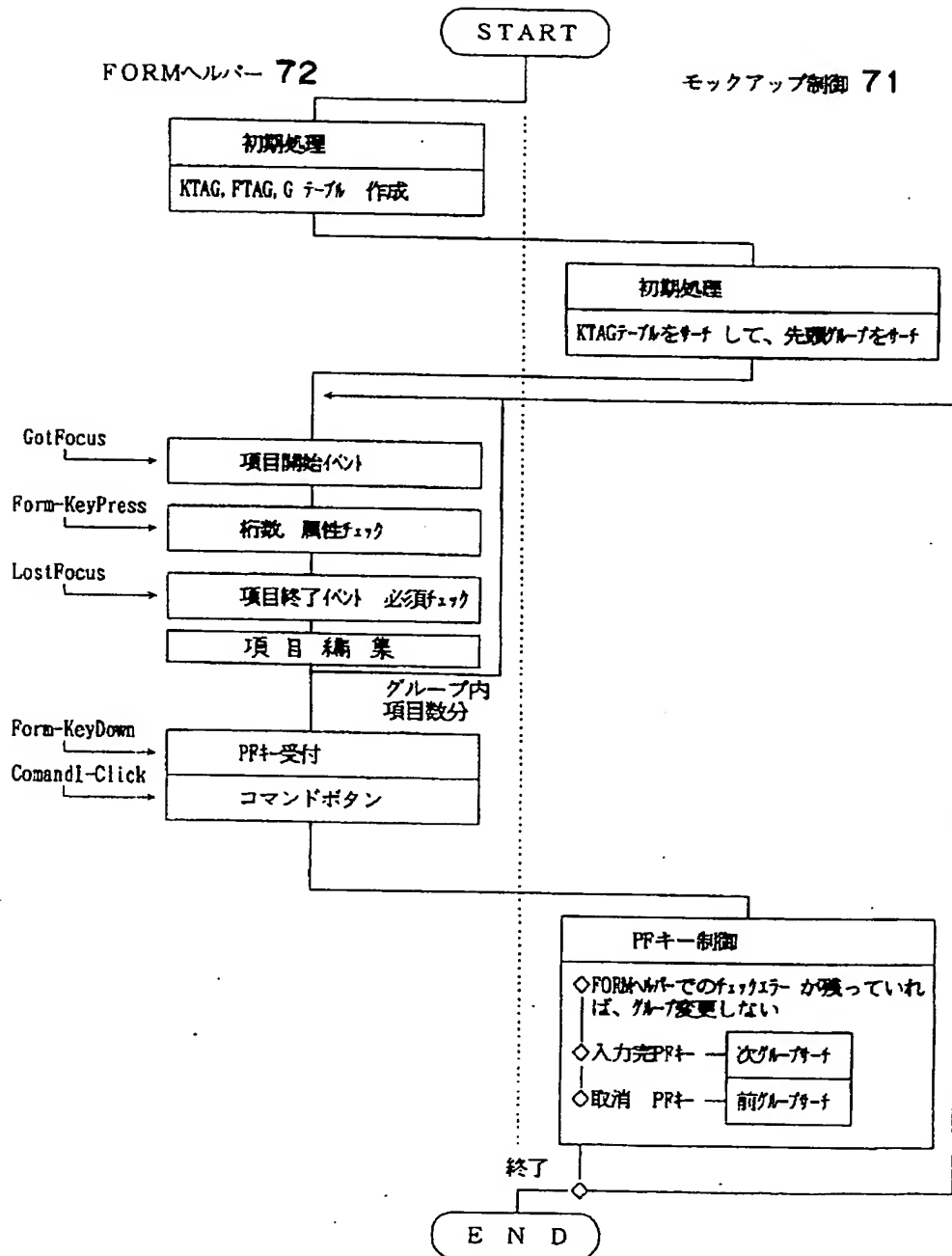
【図17】

モックアップの概略システム構造を示す図



【図18】

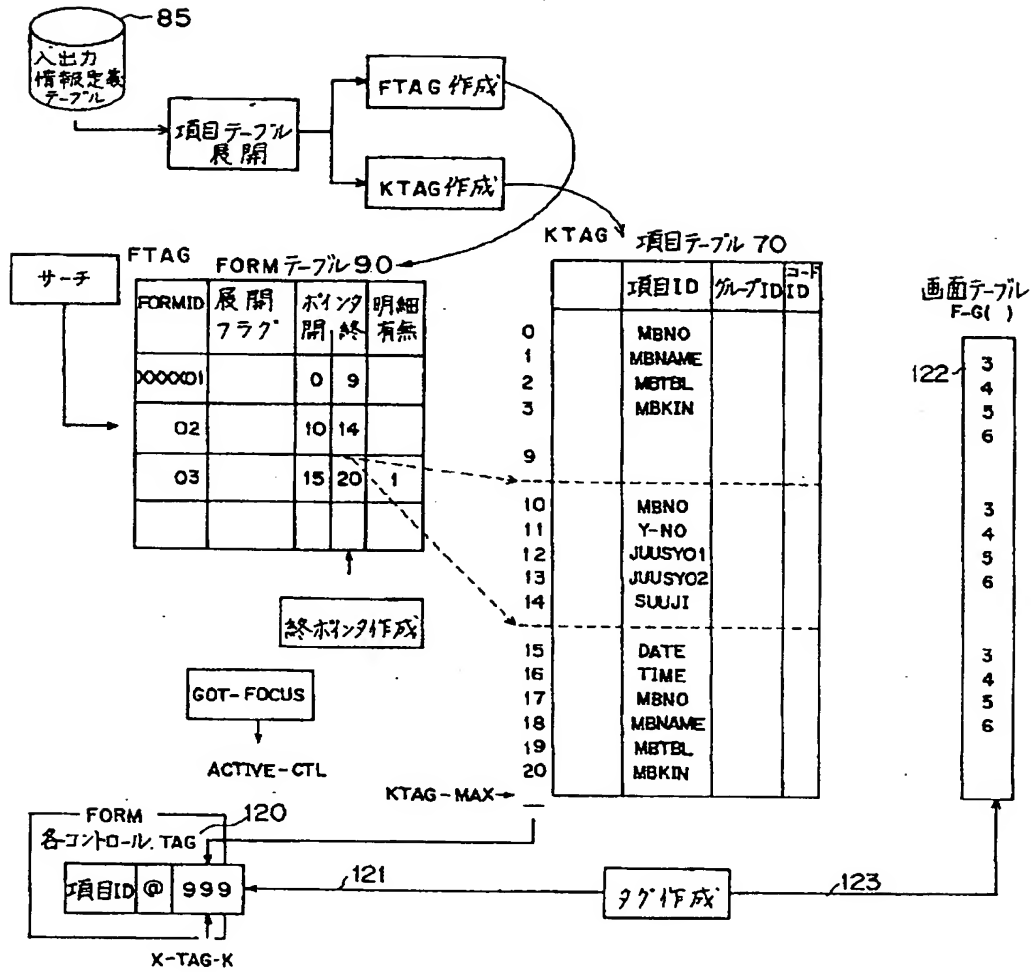
モックアップ動作のフローチャートを示す図





【図19】

## FORMヘルパー 初期処理時のテーブル構成図



【図20】

## 項目テーブル

画面ID	項目ID	桁数	データ型	グループID	必須	編集形式	項目タイプ	コ-ドID	名称表示項目
SYCENTRY	SYKGB	1	文字型	1	1	00	TXT	SYKGBUN	SYKGBNM
SYCENTRY	SYKGBNM	10	文字型	1	0	00	LBL		
SYCENTRY	SYOOD	4	文字型	2	1	00	TXT		
SYCENTRY	SYONM	20	文字型	2	1	00	LBL		
SYCENTRY	SYOKB	1	数値型	2	0	00	TXT		
SYCENTRY	SYOKIN	8	数値型	2	1	10	LBL		
KYKINPUT	KYKGB	1	文字	1	1	00	TXT		
KYKINPUT	KYONM	30	文字	1	1	00	LBL		
KYKINPUT	TAND	3	文字型	1	1	20	TXT	TANTOU	TANNM
KYKINPUT	TANNM	15	文字型	1	0	10	LBL		
KYKINPUT	KYKOMD	8	数値型	2	0	20	TXT		
KYKINPUT	KYKIN	8	数値型	2	1	10	TXT		
KYKINPUT	SYKOMD	8	数値型	2	0	20	LBL		
KYKINPUT	PRIBTN			3			BTN		

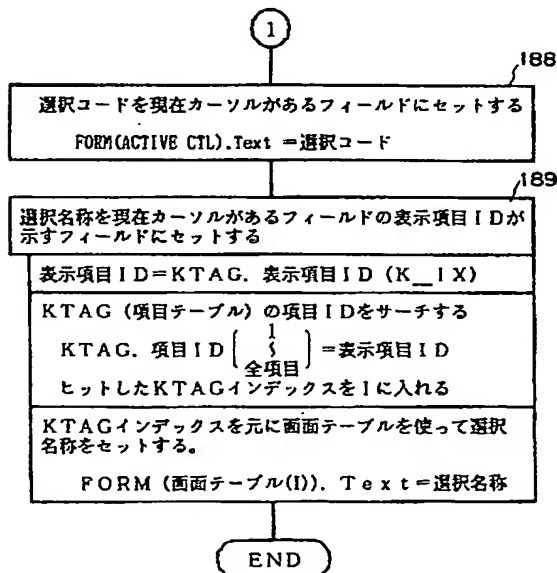
手入力情報

コード管理 手入力情報  
理由から  
選択

【図31】

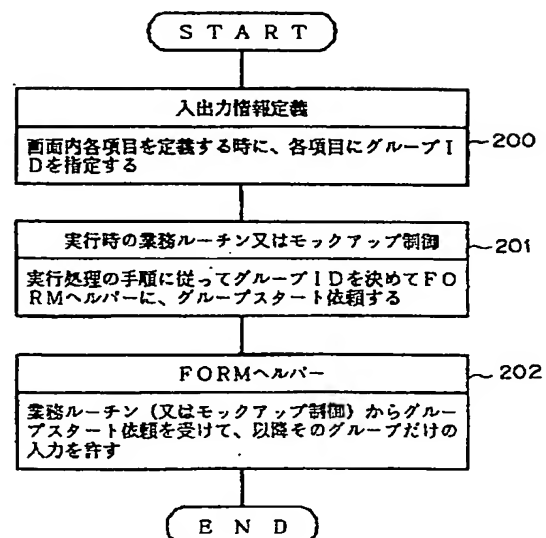
FORMヘルパーのコード参照フローチャート

(その2)



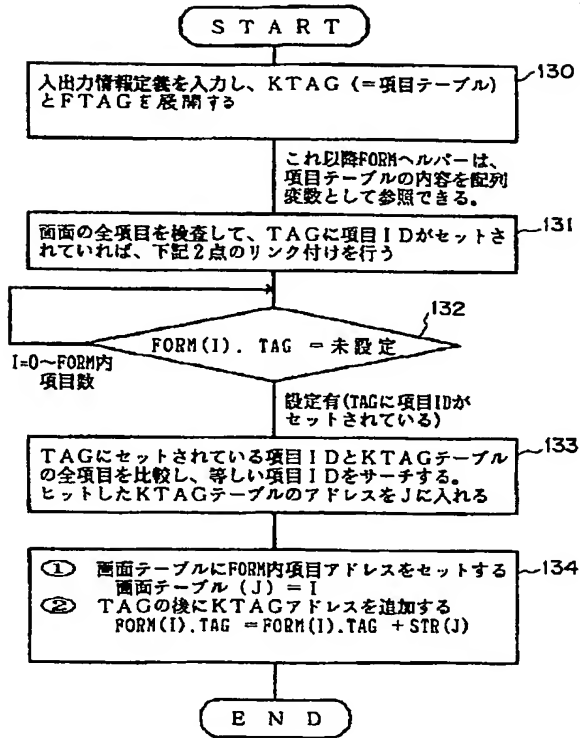
【図35】

グループ制御全体フローチャート



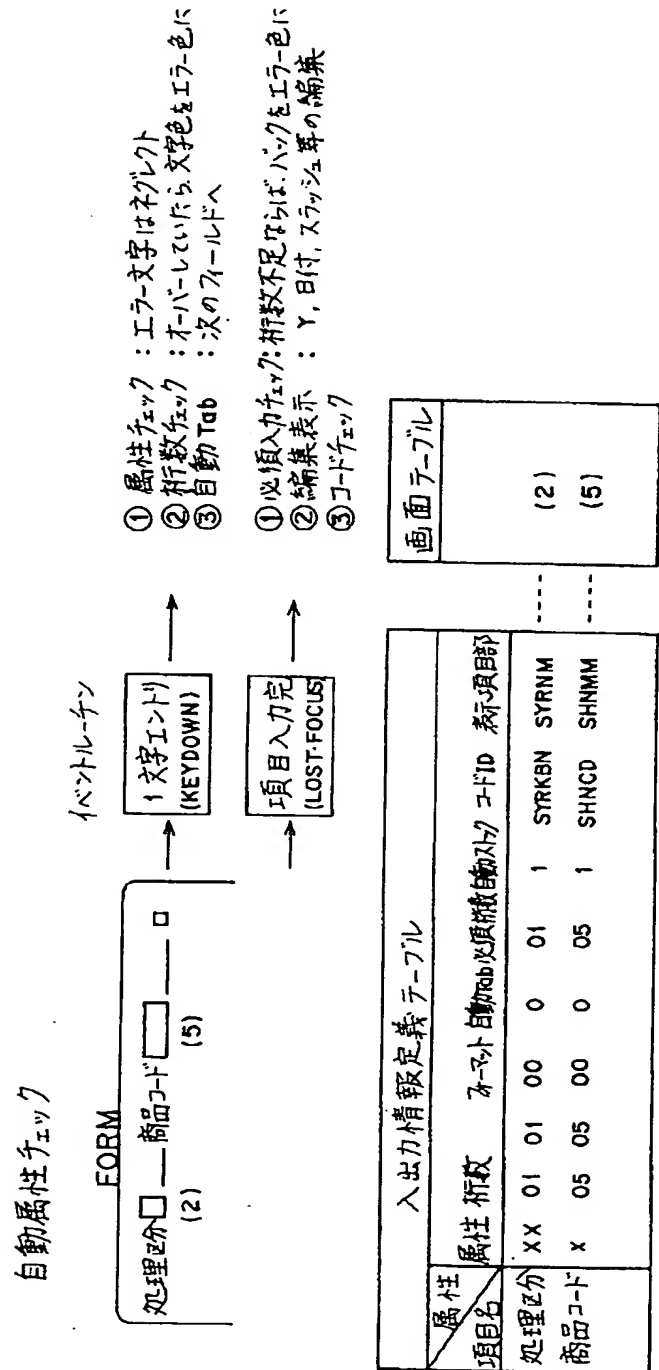
【図 21】

## FORMヘルパーの初期処理時のフローチャート



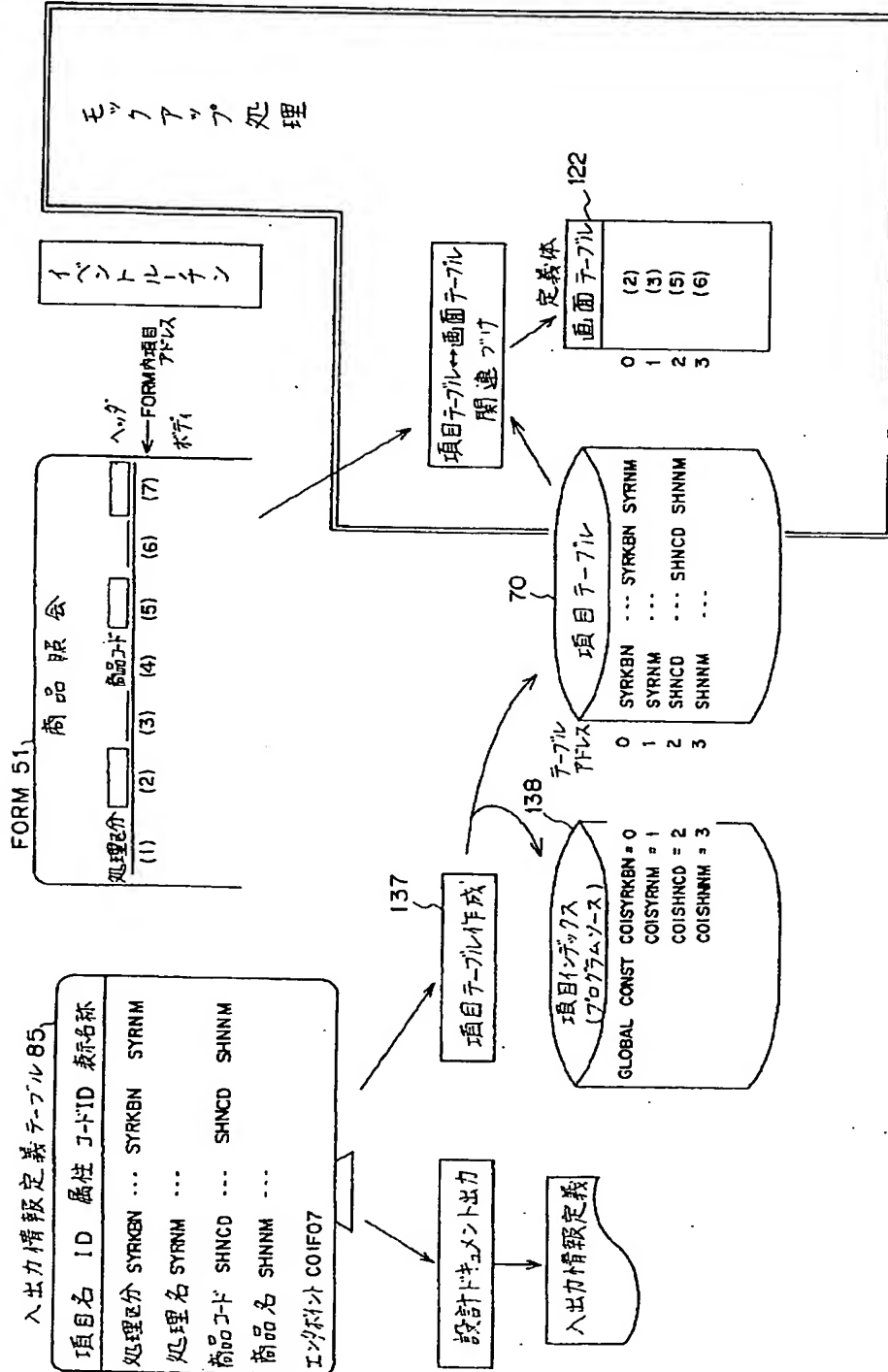
【図 23】

## 属性チェックの説明図

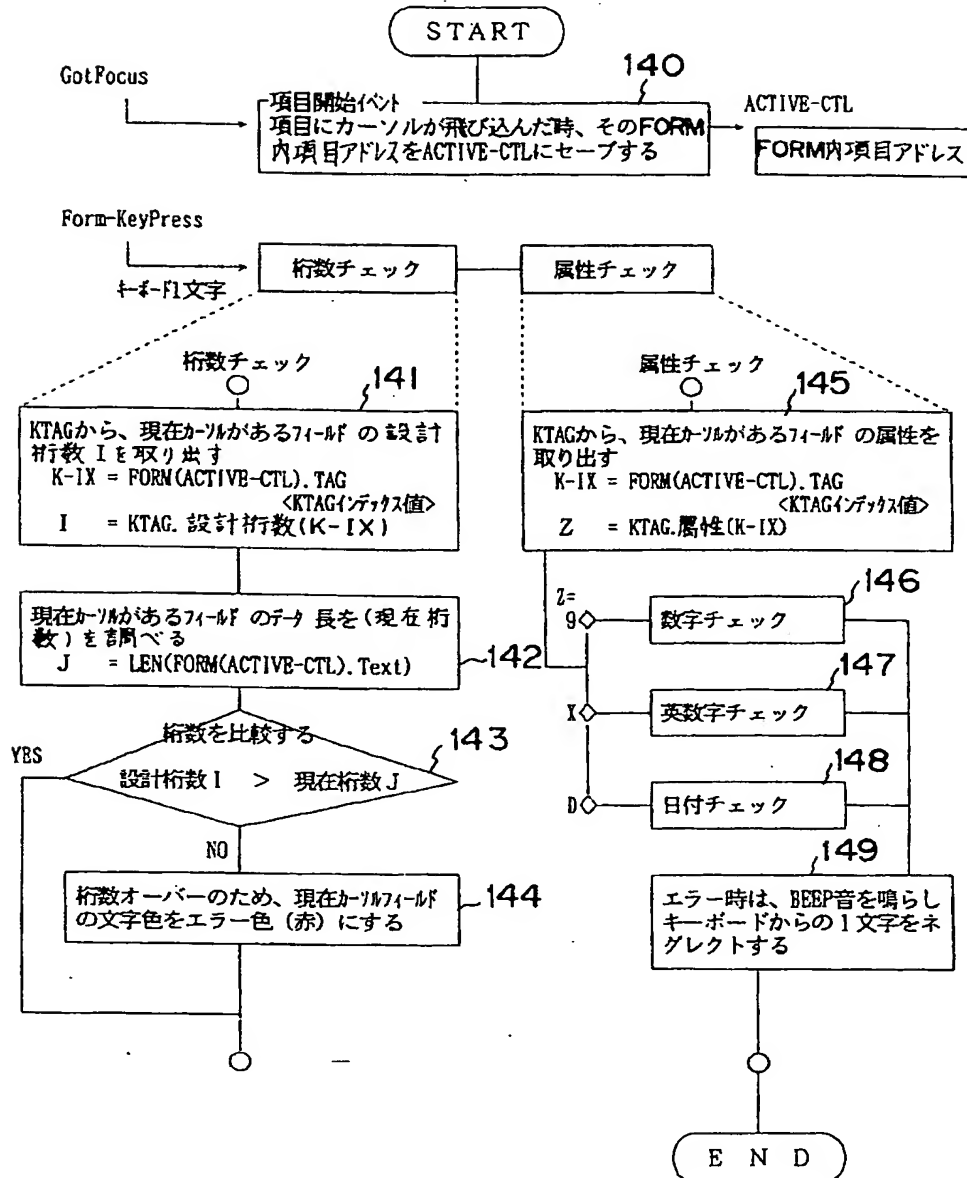


【図22】

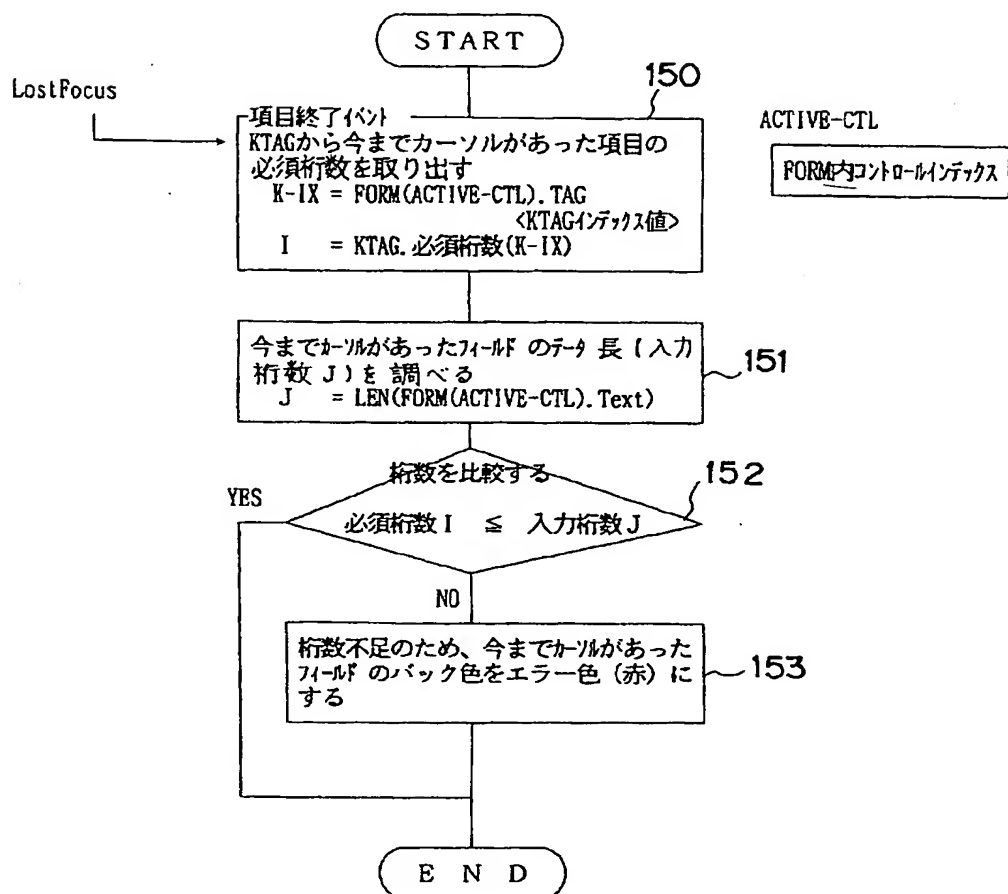
FORMヘルパーの処理を示すテーブル関係図



析数・属性チェックのフローチャートを示す図。

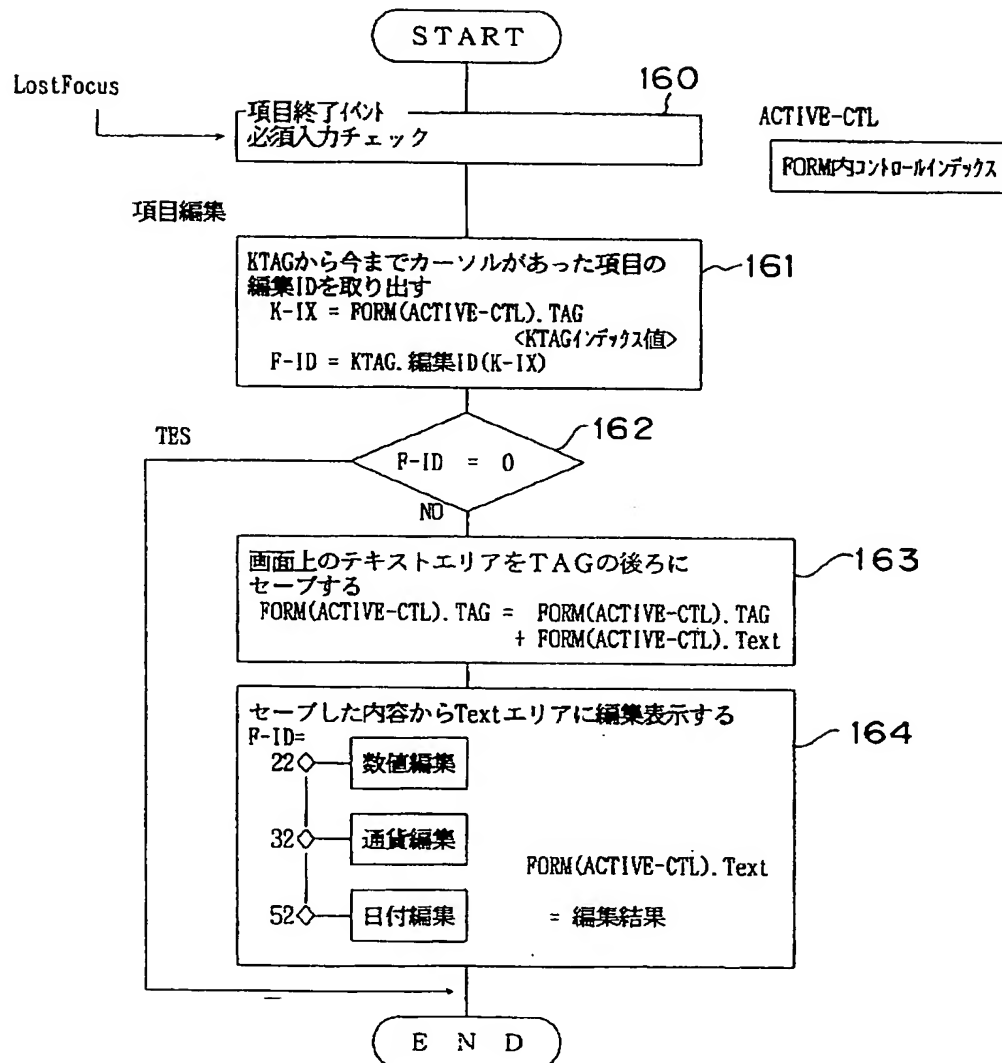


必須入力チェックのフローチャートを示す図



【図27】

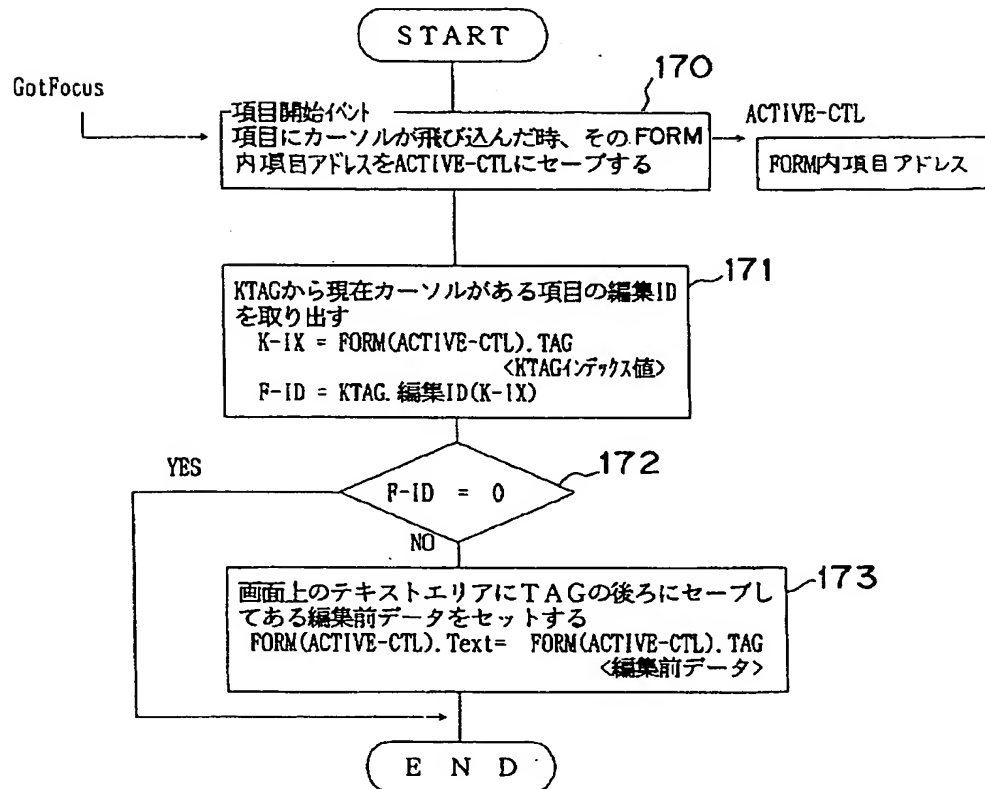
## 項目編集のフローチャートを示す図



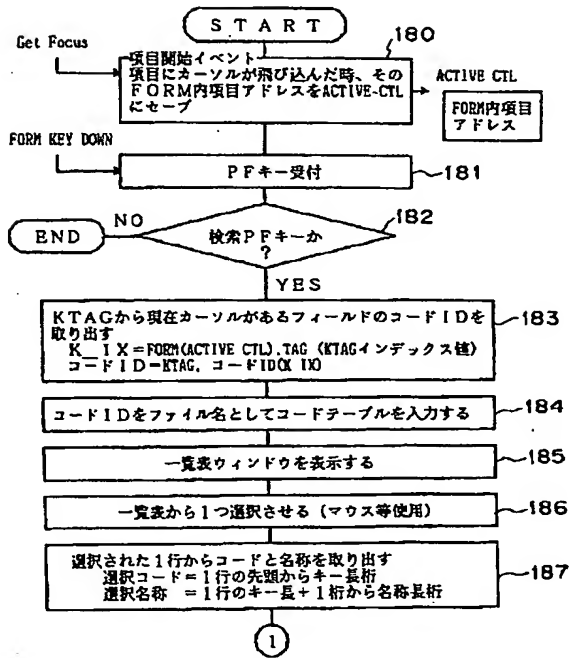


【図28】

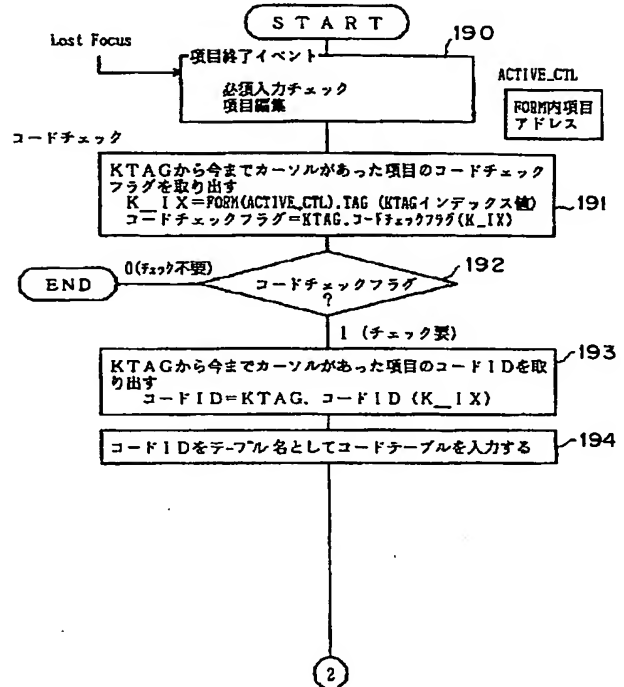
項目編集解除のフローチャートを示す図



【図30】

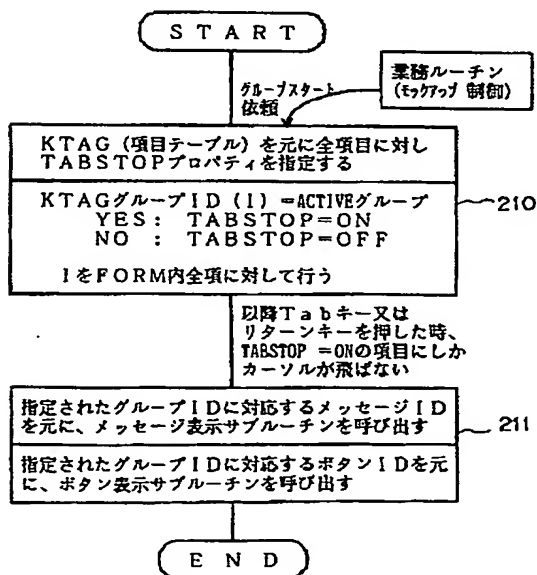
FORMヘルパーのコード参照フローチャート  
(その1)

【図32】

FORMヘルパーのコードチェックフローチャート  
(その1)

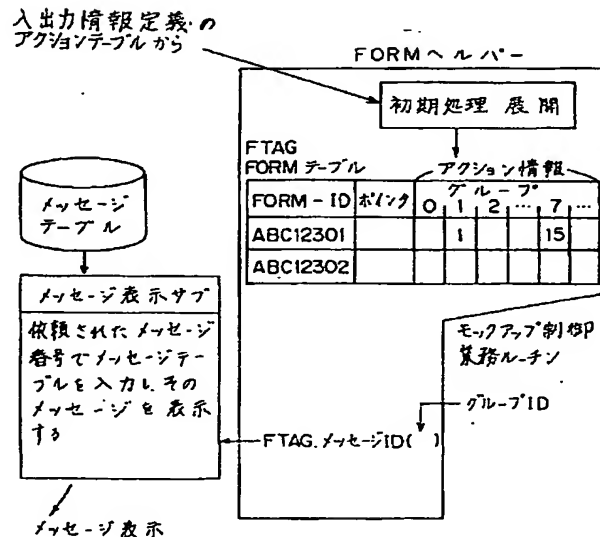
【図36】

## FORMヘルパーのグループ制御フローチャート

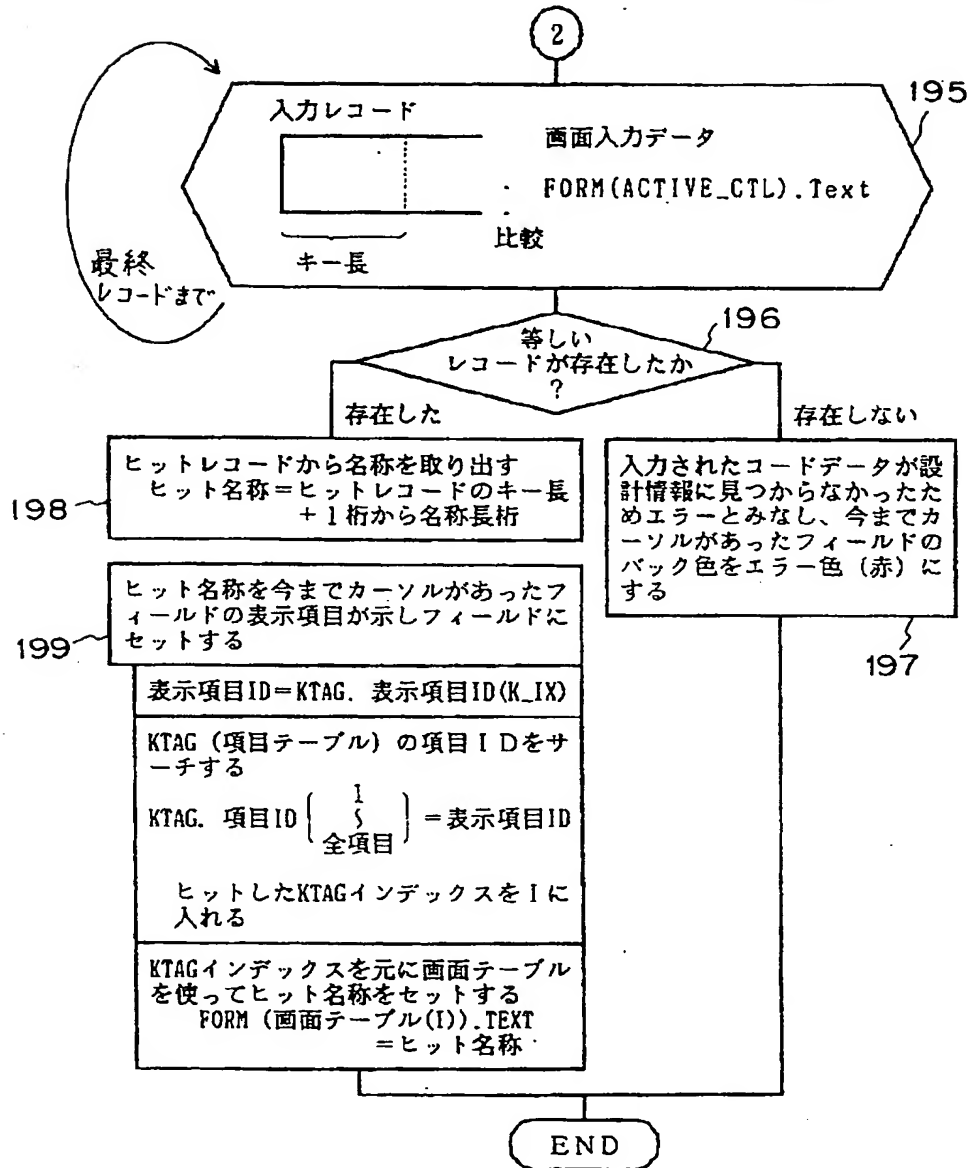


【図37】

## 入出力情報定義のアクション動作の説明図

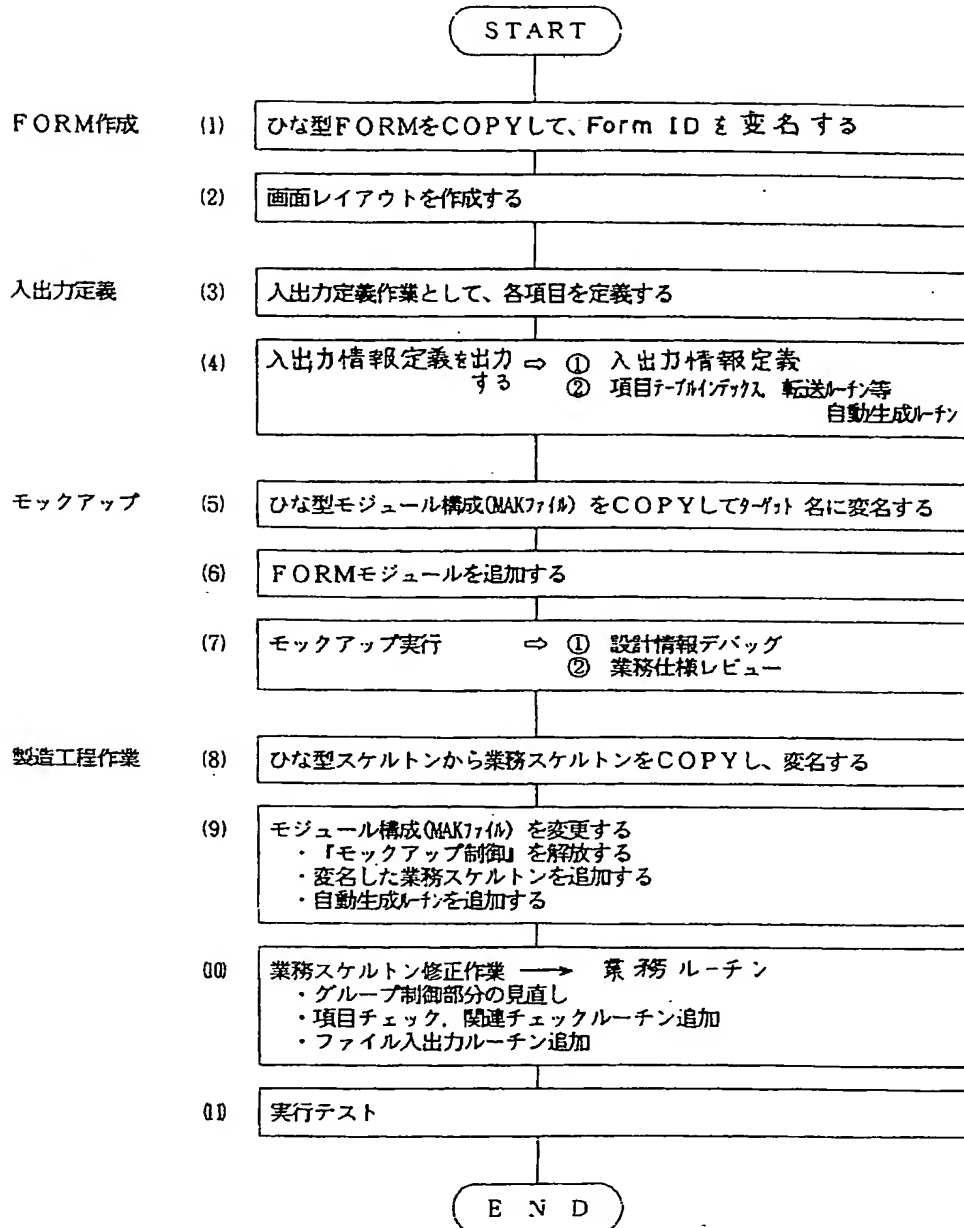


【図33】

FORMヘルパーのコードチェックフローチャート  
(その2)

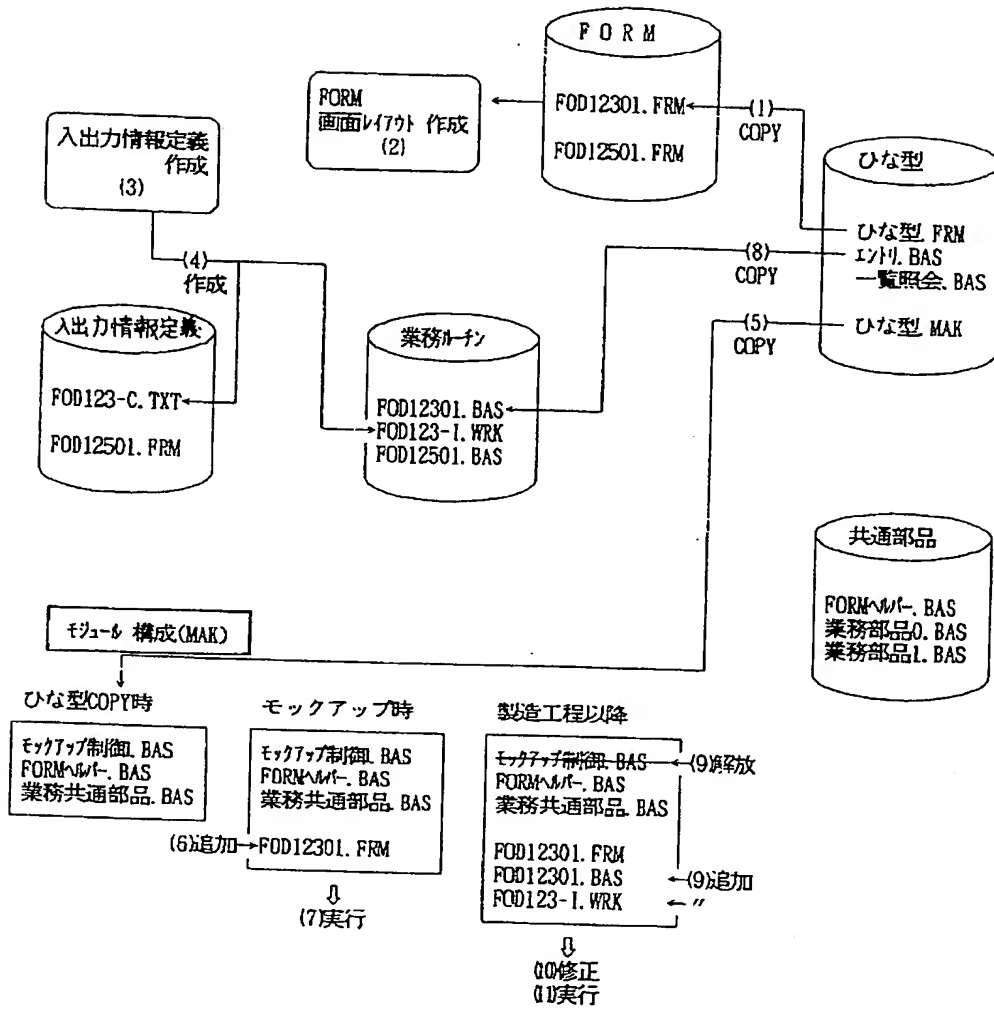
【図38】

## 部品資産継承 フローチャート



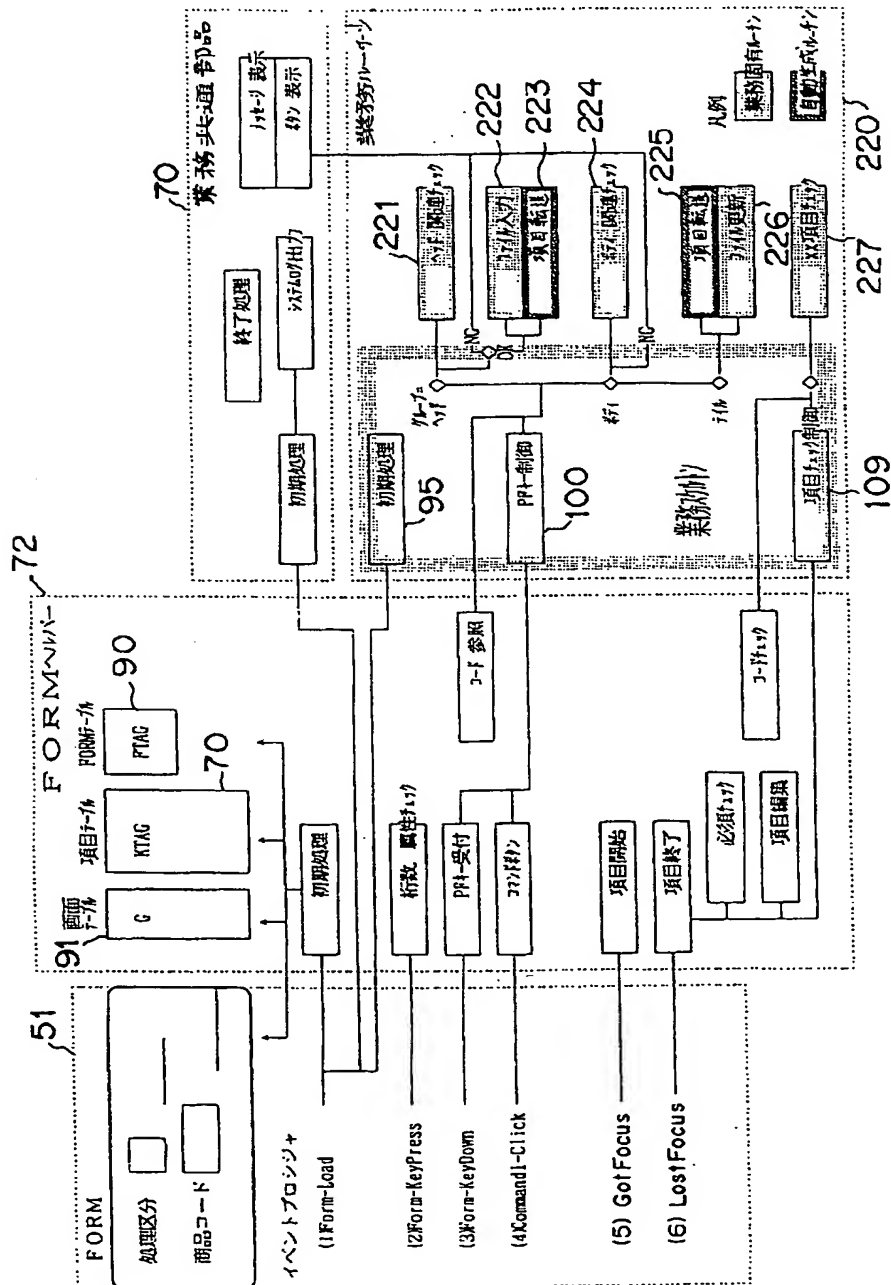
【図 39】

## 部品継承を説明するためのブロック図



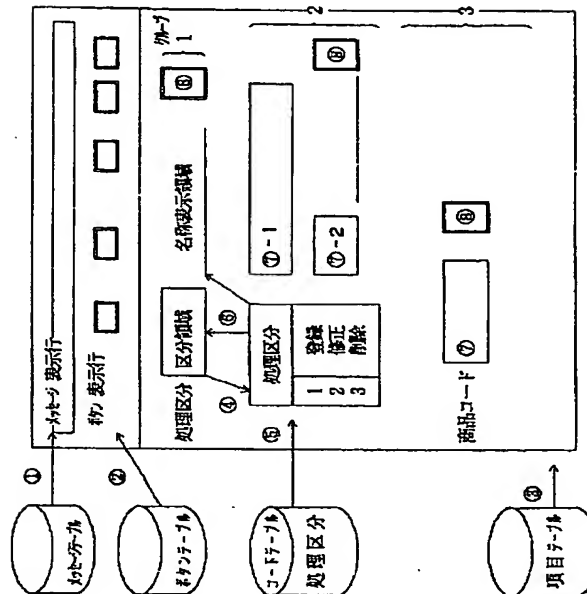
【図40】

FORM, FORMヘルパー 業務共通部品を  
継承して製造した業務ルーチンアプリケーションの構成図



【図41】

本発明によるモックアップ動作および  
これを継承した業務ルーチンアプリケーション動作を  
表示するディスプレイ図



フロントページの続き

(72)発明者 木村 隆行  
東京都港区西新橋三丁目25番33号 株式会  
社富士通東京システムズ内

(72)発明者 上岡 靖  
東京都港区西新橋三丁目25番33号 株式会  
社富士通東京システムズ内

(72)発明者 加藤 健二  
東京都港区西新橋三丁目25番33号 株式会  
社富士通東京システムズ内